## Contents

**ServiceNow Platform**

<table>
<thead>
<tr>
<th>ServiceNow Platform</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation and the user interface</td>
<td>6</td>
</tr>
<tr>
<td>Activate UI16</td>
<td>6</td>
</tr>
<tr>
<td>Logging in</td>
<td>10</td>
</tr>
<tr>
<td>Generally supported browsers</td>
<td>10</td>
</tr>
<tr>
<td>Switch between UI16 and UI15</td>
<td>11</td>
</tr>
<tr>
<td>Compare UI16 and UI15 styles</td>
<td>12</td>
</tr>
<tr>
<td>Configure roles allowed to switch between UI16 and UI15</td>
<td>15</td>
</tr>
<tr>
<td>Switch between UI15 and UI11</td>
<td>15</td>
</tr>
<tr>
<td>Installed components for UI plugins</td>
<td>16</td>
</tr>
<tr>
<td>Keyboard shortcuts</td>
<td>17</td>
</tr>
<tr>
<td>UI16 banner frame</td>
<td>18</td>
</tr>
<tr>
<td>UI15 banner frame</td>
<td>21</td>
</tr>
<tr>
<td>UI11 banner frame</td>
<td>26</td>
</tr>
<tr>
<td>The Edge</td>
<td>27</td>
</tr>
<tr>
<td>Application navigator</td>
<td>31</td>
</tr>
<tr>
<td>URL schema</td>
<td>53</td>
</tr>
<tr>
<td>Lists</td>
<td>56</td>
</tr>
<tr>
<td>Forms</td>
<td>96</td>
</tr>
<tr>
<td>User presence</td>
<td>135</td>
</tr>
<tr>
<td>Common UI elements</td>
<td>138</td>
</tr>
<tr>
<td>Tags</td>
<td>166</td>
</tr>
</tbody>
</table>

**User interface configuration**

<table>
<thead>
<tr>
<th>User interface configuration</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation stack</td>
<td>177</td>
</tr>
<tr>
<td>Navigation action cancellation</td>
<td>180</td>
</tr>
<tr>
<td>Context-sensitive help</td>
<td>186</td>
</tr>
<tr>
<td>System user guide</td>
<td>192</td>
</tr>
<tr>
<td>Creating a change password module</td>
<td>201</td>
</tr>
<tr>
<td>Define locations</td>
<td>202</td>
</tr>
<tr>
<td>Database storage for audio files</td>
<td>204</td>
</tr>
<tr>
<td>Storing images in the database</td>
<td>205</td>
</tr>
<tr>
<td>User preferences</td>
<td>210</td>
</tr>
<tr>
<td>View management</td>
<td>234</td>
</tr>
<tr>
<td>Welcome page content</td>
<td>241</td>
</tr>
<tr>
<td>Google Maps integration</td>
<td>243</td>
</tr>
<tr>
<td>Map pages</td>
<td>246</td>
</tr>
<tr>
<td>Theming an instance</td>
<td>257</td>
</tr>
<tr>
<td>Create a company profile</td>
<td>271</td>
</tr>
<tr>
<td>Comparing field values</td>
<td>280</td>
</tr>
<tr>
<td>Add help to a field label on a form</td>
<td>284</td>
</tr>
<tr>
<td>Context menus</td>
<td>287</td>
</tr>
</tbody>
</table>

**Administer the ServiceNow Platform**

<table>
<thead>
<tr>
<th>Administer the ServiceNow Platform</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service administration</td>
<td>295</td>
</tr>
<tr>
<td>List administration</td>
<td>296</td>
</tr>
<tr>
<td>Form administration</td>
<td>672</td>
</tr>
<tr>
<td>Field administration</td>
<td>703</td>
</tr>
<tr>
<td>Currency administration</td>
<td>777</td>
</tr>
<tr>
<td>Localization</td>
<td>955</td>
</tr>
<tr>
<td>Time configuration</td>
<td>964</td>
</tr>
<tr>
<td>Search administration</td>
<td>1066</td>
</tr>
</tbody>
</table>
Mobile device configuration......................................................... 1115
Core configuration........................................................................... 1233
Data management......................................................................... 1435
Integration with third-party applications and data sources........ 1684
User administration...................................................................... 1722
Upgrades and conversions............................................................. 1969
Platform performance................................................................... 1995
Content Management System......................................................... 2040
Content Management System design.............................................. 2041
Activate the Content Management System................................. 2048
Configure Content Management sites.......................................... 2050
Content Management integration points........................................ 2111
Content Management testing......................................................... 2120
Content Management administration............................................. 2120
Global search in Content Management.......................................... 2121
CMS translation............................................................................ 2121
Knowledge Management................................................................ 2123
Knowledge Management setup guide for admins......................... 2124
Knowledge base setup guide for knowledge admins and managers 2126
Knowledge Management guide for users....................................... 2127
Knowledge Management roles....................................................... 2128
Knowledge homepage................................................................. 2129
Social Q&A.................................................................................... 2132
Knowledge feedback...................................................................... 2137
Request a knowledge base............................................................. 2138
Create a knowledge article............................................................. 2138
Knowledge manager..................................................................... 2143
Knowledge administration............................................................. 2153
Use knowledge on mobile devices................................................. 2166
Knowledge Management v3 migration.......................................... 2169
Dependency Views........................................................................ 2175
Installed with Dependency Views.................................................. 2175
Dependency Views map................................................................. 2180
Dependency Views map menus and controls............................... 2181
Supported browsers for Dependency Views................................. 2184
Collapsed nodes in a Dependency Views map............................. 2184
Use Dependency Views................................................................. 2185
Administer Dependency Views...................................................... 2189
Legacy Business Service Management Map................................. 2194
Visual Task Boards....................................................................... 2231
Supported browsers for Visual Task Boards................................. 2232
Board structure............................................................................. 2232
Activate Visual Task Boards......................................................... 2236
Installed with Visual Task Boards.................................................. 2236
Visual Task Board use................................................................. 2238
Configure the task board............................................................... 2258
Configure Visual Task Board labels.............................................. 2260
Configure the card limit for freeform boards............................... 2261
Configure the card limit for flexible and guided boards................. 2261
Connect......................................................................................... 2261
Supported browsers for Connect.................................................... 2262
Activate Connect.......................................................................... 2262
Activate Connect Support.............................................................. 2265
Connect interface.......................................................................... 2267
Enable or disable Connect notifications globally.......................... 2277
Edit which notifications you receive for a conversation............... 2278
ServiceNow Platform

The ServiceNow Platform is a powerful cloud application platform that enables you to link real-time data with activities, tasks, and processes to achieve better work outcomes.

Navigation and the user interface

The user interface (UI) is the main way to interact with the applications and information in a ServiceNow instance.

The following versions of the user interface are available. Each new version is intended to provide an updated look and usability improvements.

UI16

Notable features of the UI16 interface include real-time form updates, user presence, a redesigned application navigator with tabs for favorites and history, and enhanced activity streams. UI16 is the default user interface for new instances. For upgraded instances, an administrator must activate UI16.
UI15

Features introduced in the UI15 interface included redesigned icons, the ability to specify when related lists load with a form and other new options on the system menu, and form personalization.

Figure 2: UI15 user interface

UI11

UI11 builds on the classic user interface by adding the Edge and several features including split screen, bookmarks, and flyout windows. UI11 is activated but disabled by default.
Figure 3: UI11 user interface

Table 1: User interface components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banner frame</td>
<td>Displays at the top of every page and contains a logo, global navigation controls, and tools. See <em>UI16 banner frame</em> on page 18, <em>UI15 banner frame</em> on page 21, or <em>UI11 banner frame</em> on page 26.</td>
</tr>
<tr>
<td>Application navigator</td>
<td>Also called the left-navigation bar. Provides links to all applications and modules. See <em>UI16 application navigator</em> on page 31, <em>UI15 application navigator</em> on page 38, or <em>UI11 application navigator</em> on page 40.</td>
</tr>
<tr>
<td>Content frame</td>
<td>Displays information such as lists, forms, homepages, and wizards.</td>
</tr>
<tr>
<td>The Edge</td>
<td>A toolbar on the left side of the UI15 and UI11 interfaces that provides quick links to commonly used features. See <em>The Edge</em> on page 27.</td>
</tr>
<tr>
<td></td>
<td>In UI16, the Edge is replaced by the <em>Favorites</em> tab of the application navigator represented by a star. The collapsed view of the UI16 application navigator is similar to the Edge.</td>
</tr>
</tbody>
</table>
Activate UI16

UI16 is the default user interface for new instances. For upgraded instances, you can activate the UI16 plugin (com.glide.ui.ui16) if you have the admin role.

Confirm that the instance upgrade completed successfully by visiting the Upgrade Monitor page.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Logging in

Each ServiceNow instance has a unique, secure web address. The base URL for each instance has the default format: https://<instancename>.service-now.com

Role required: none

Users log in to the instance from a web browser. Administrators can create a custom URL that redirects to the assigned URL, if needed.

1. Enter the base URL in any web browser.
   - If your system uses external authentication, you are automatically logged in. For example, you may log in to company services when you log in to your computer.
   - If your system does not use external authentication, the Welcome page appears.
2. Enter your user name and password.
3. Optional: Select the Remember Me check box to remain logged in until you manually log out.
   - This option can be enabled or disabled by the administrator. For more information, see Remember me login cookie on page 2457.
4. Press the Enter key or click Login.

Generally supported browsers

Browser support varies for each version of the user interface (UI). Most major browsers are supported.

Note: Some features have additional browser requirements, which are noted in the appropriate documentation.
Table 2: Browser support for each UI version

<table>
<thead>
<tr>
<th>UI version</th>
<th>Chrome support</th>
<th>Firefox support</th>
<th>Internet Explorer support</th>
<th>Microsoft Edge support</th>
<th>Safari support</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI16</td>
<td>Latest public release</td>
<td>Latest public release of Firefox or Firefox ESR</td>
<td>9 and up • Edge mode is supported</td>
<td>Latest public release</td>
<td>6.1 and up</td>
</tr>
<tr>
<td>UI15</td>
<td>Latest public release</td>
<td>Latest public release of Firefox or Firefox ESR</td>
<td>9 and up • Edge mode is supported</td>
<td>Latest public release</td>
<td>6.1 and up</td>
</tr>
<tr>
<td>UI11</td>
<td>Latest public release</td>
<td>Latest public release of Firefox or Firefox ESR</td>
<td>7 and up • Edge mode is not supported</td>
<td>Latest public release</td>
<td>6 and up</td>
</tr>
</tbody>
</table>

**Internet Explorer notes**

- Compatibility mode is not supported.
- Setting Security Mode to High (via Internet Options > Security tab) is not supported.
- While UI16 and UI15 are supported on Internet Explorer 9, the browser may be slow when performing some operations. This does not necessarily indicate a performance degradation in the instance.
- Internet Explorer 11 is susceptible to memory leaks, which may impact performance, especially in Windows 7.

**Safari notes**

- For Safari version 5 and Internet Explorer versions 7 and 8, the user interface automatically redirects to UI11.
- There is a security issue in some versions of Safari. User credentials may be disclosed to an unexpected site via autofill. For more information, see the Apple security update.
- Starting with Safari 9.1 and future versions of Safari, the leave page / stay on page pop-up message will only display once.

**Switch between UI16 and UI15**

You can switch between the UI16 and UI15 user interface versions.

Role required: admin
After UI16 is activated, all users see the enhanced UI16 user interface. By default, administrators see a button at the top of the System Settings window that lets them switch between UI16 and UI15. The administrator can add a system property to configure other roles that see the button.

1. Click the gear icon ( ) in the banner frame to open the System Settings window.

2. Click **Switch to UI15** or **Switch to UI16**.
   The button that displays is based on which UI version you are using when you open System Settings. The application refreshes with the selected UI. Your homepage appears in the content frame. You can access your favorites on the Edge (UI15) or in the favorites tab of the application navigator (UI16).

Comparison of UI16 and UI15 styles

Review the basic style differences between UI16 and UI15 styles in the ServiceNow platform.

**Banner frame comparison**

In UI16, the banner information and controls are in different places. The search box in the UI16 header opens when you select it. The help icon is available from the banner frame, instead of being accessed from the Settings menu. Because the UI16 banner is narrower, there is no control to collapse it.

![Figure 4: Banner frame differences](image)

For more information about the UI16 changes, see **UI16 banner frame** on page 18.
Application navigator

In UI16, modules that you select are not automatically marked as favorites as illustrated by the blue stars in the UI15 image. Any menu or module can be marked as a favorite, and it appears when you click the Favorites tab represented by the star. The other tab displays items you recently accessed.

For more information, see UI16 application navigator on page 31. The navigation filter works the same in both UI versions, by entering text in the navigation filter to show matching applications, modules, and favorites.

Another difference in the UI16 navigator menu is that you can no longer right-click an application menu name to edit the application menu. You must navigate to System Definition > Application Menus to add or modify modules.

The Edge

The Edge in UI15 is replaced in UI16 by the Favorites tab, represented by the star. Any item you could place on the Edge can be marked as a favorite. When you switch to UI16, all bookmarks on the edge are migrated to favorites and are listed on the Favorites tab. For more information, see Use favorites in UI16 on page 34. The List and Form View option is no longer available in UI16.
System settings

UI16 System Settings appear in a pop-up window with subtabs to make it easier to update your settings. The Split Layout setting available in UI15 was removed. For more information, see *UI16 banner frame* on page 18.

![System Settings](image)

Figure 5: UI16 System Settings
Basic configuration and the company logo

As the banner size and color scheme has changed significantly in UI16, there is a set of system properties for UI16 to set up your logo and branding before you switch to UI16. For more information, see Configure logo, colors, and system defaults for UI16 on page 257.

The banner in which your logo appears is narrow, and your logo image is scaled based on the aspect ratio. A wider image scales better than a smaller one and is more readable, as in the following example. It is also recommended that you create a logo image with a transparent background.

![Square logo](image1)

![Wide logo](image2)

Figure 6: Square logo  
Figure 7: Wide logo

Configure roles allowed to switch between UI16 and UI15

After you enable UI16, users with the admin role see a button in the System Settings window that allows them to switch between UI16 and UI15. You can add a system property to configure other roles that are allowed to switch UI versions.

Role required: admin

You may need to allow users who develop or configure applications to view the changes in both versions of the user interface. You list the roles in the new system property, and users with one of the roles see the switch button at the top of the System Settings window.

1. Add a new system property with the following field values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.ui.ui15_switch_roles</td>
</tr>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td>Enter a comma-separated list of roles that are allowed to switch between UI16 and UI15. If you set the value to Public, all users will be able to switch between the UI versions.</td>
</tr>
</tbody>
</table>

2. Click Submit.

Test the new system property by impersonating a user with one of the roles and verifying that the switch button appears when the user opens the System Settings window.

Switch between UI15 and UI11

You can disable UI15, which reverts the user interface to UI11.

If UI16 is enabled, you must first switch to UI15.

Role required: admin

1. Enter `sys_properties.list` in the navigation filter and press the Enter key.
2. Locate the glide.ui.doctype property in the System Properties list. The property is set to true by default, which enables UI15.

3. Set the property Value to false.

4. Log out and back in. This disables UI15 for the instance. The interface reverts to UI11 in individual user sessions when each user logs out and logs back in.

Installed components for UI plugins

The following components are added with the user interface plugins.

Tables

Table 3: User interface tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmarks [sys_ui_bookmark]</td>
<td>Stores user bookmarks and preferences. To access, navigate to System Definition &gt; Bookmarks.</td>
</tr>
</tbody>
</table>

Properties

Table 4: User interface properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.allow_ie_compatibility_mode</td>
<td>In many cases, Internet Explorer is configured to use compatibility mode when viewing ServiceNow. However, UI15 does not display when compatibility mode is turned on. When this property is set to false, the system does not allow use of compatibility mode on UI15 pages. The default value for this property is false.</td>
</tr>
<tr>
<td>glide.ui.doctype</td>
<td>Enables or disables the UI15 interface. The default value for this property is true. After enabling or disabling this property, the administrator must log out and log back in to effect the change in the instance. When users log out and log back in, they see the change in their individual user sessions.</td>
</tr>
<tr>
<td>glide.ui.doctype.ie_version</td>
<td>Sets the minimum version of Internet Explorer required to use UI15. The default value for this property is 9. It is recommended that you do not change this value.</td>
</tr>
</tbody>
</table>
User preferences

Table 5: User interface user preferences

<table>
<thead>
<tr>
<th>User Preference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.navpage_state</td>
<td>Do not edit this preference; system record of a user's UI11 configuration.</td>
</tr>
<tr>
<td>ng.navigator_preference</td>
<td>Stores a user's selected application navigator favorites in UI15.</td>
</tr>
<tr>
<td>glide.ui.field_style_circles</td>
<td>Shows field status indicators on lists as circles instead of cell backgrounds. The system administrator can set this preference for new logins. Users can make this change by navigating to a list, clicking the personalize list icon (/modal), and selecting the Modern cell coloring check box.</td>
</tr>
</tbody>
</table>

Keyboard shortcuts

You can use keyboard shortcuts to quickly perform common actions in the user interface.

Access keys depend on the browser and operating system you are using. Available keyboard shortcuts are based on the UI version.

Table 6: UI16 keyboard shortcuts

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate global search field</td>
<td>Access key + G</td>
</tr>
<tr>
<td>Toggle application navigator</td>
<td>Access key + C</td>
</tr>
<tr>
<td>Activate navigation filter field</td>
<td>Access key + F</td>
</tr>
<tr>
<td>Impersonate user</td>
<td>Access key + I</td>
</tr>
</tbody>
</table>

Table 7: UI15 and UI11 keyboard shortcuts

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard shortcut</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate global search field</td>
<td>Access key + S</td>
<td></td>
</tr>
<tr>
<td>Toggle banner frame</td>
<td>Access key + B</td>
<td>Ensure the page focus is on the content frame</td>
</tr>
<tr>
<td>Toggle application navigator</td>
<td>Access key + N</td>
<td></td>
</tr>
<tr>
<td>Activate navigation filter field</td>
<td>Access key + F</td>
<td></td>
</tr>
<tr>
<td>Toggle list and form view</td>
<td>Access key + V</td>
<td></td>
</tr>
<tr>
<td>Toggle horizontal split</td>
<td>Access key + H</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard shortcut</td>
<td>Notes</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximize the current pane</td>
<td>Access key + M</td>
<td></td>
</tr>
<tr>
<td>Send email</td>
<td>Access key + S</td>
<td></td>
</tr>
<tr>
<td>Submit form</td>
<td>Enter</td>
<td></td>
</tr>
</tbody>
</table>

### UI16 banner frame

The banner frame runs across the top of every page.

The banner frame for UI16 contains a logo and the following information, controls, and tools. To configure the banner frame logo, text, and colors, see *Configure logo, colors, and system defaults for UI16* on page 257.

#### Table 8: UI16 banner frame controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User menu</td>
<td>Provides access to the following options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Profile</strong>: Opens your user profile, where you can edit personal details and customize your notification preferences.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Impersonate User</strong>: Allows users with the impersonator role to switch user views without logging out; useful for testing security and role-specific setup functions.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Elevate Roles</strong>: Allows users with elevated privilege roles to elevate to a higher security role when needed and then return to their normal role.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Logout</strong>: Returns to the Welcome page for subsequent login.</td>
</tr>
<tr>
<td>Connect sidebar icon</td>
<td>Opens the Connect sidebar to begin or continue conversations.</td>
</tr>
<tr>
<td>Global text search icon</td>
<td>Searches for text in multiple applications.</td>
</tr>
<tr>
<td>Help icon</td>
<td>Provides options to tour what is new in UI16 and to open the User Guide or Product Documentation.</td>
</tr>
<tr>
<td></td>
<td>• <strong>What's New</strong>: Opens a short slideshow that highlights new UI16 features.</td>
</tr>
<tr>
<td></td>
<td>• <strong>User Guide</strong>: Opens the user guide, an in-product set of documents designed to introduce users to the platform. This option is available only when the User Guide plugin is activated.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Search Product Documentation</strong>: Opens the official ServiceNow product documentation.</td>
</tr>
</tbody>
</table>
The gear icon, located within the banner frame, displays the system settings pop-up window. The system settings are organized by tab. At the top, administrators (admin role) see the **Switch to UI15** button if UI16 is activated.

## Table 9: System settings

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gear icon (⚙️)</td>
<td>Displays the system menu with additional settings and controls.</td>
</tr>
<tr>
<td><strong>General tab</strong></td>
<td></td>
</tr>
<tr>
<td>Accessibility Enabled</td>
<td>Section 508 accessibility features are activated when this setting is enabled.</td>
</tr>
<tr>
<td>Compacts the user interface</td>
<td>The UI is optimized to display more information in the browser window when this setting is enabled.</td>
</tr>
<tr>
<td>Wrap Longer Text in List Columns</td>
<td>Long strings wrap in list columns instead of appearing as one long line when this setting is enabled.</td>
</tr>
<tr>
<td>Compact list date/time</td>
<td>Date and time values appear in a compact format when this setting is enabled. The year is not shown for date values within the current year, and seconds are not shown for time values. This setting is not available if the <strong>Date/Time</strong> selection is <strong>Time Ago</strong>.</td>
</tr>
<tr>
<td><strong>Date/Time</strong></td>
<td>The format in which dates and times appear depends on this setting. Select <strong>Calendar</strong> (for example, 2015-11-13, 15:58:58), <strong>Time Ago</strong> (for example, 11 minutes ago), or <strong>Both</strong>.</td>
</tr>
<tr>
<td>Language</td>
<td>Select a language or return to the default language (🌐). This setting is available if a <strong>language plugin</strong> is activated.</td>
</tr>
<tr>
<td>Time zone</td>
<td>Select a time zone or return to the default time zone (🕒).</td>
</tr>
<tr>
<td><strong>Domain</strong></td>
<td>Specifies the currently selected domain for organizations that are using domain separation. The user must have the domain_admin or admin role for this setting to appear.</td>
</tr>
<tr>
<td>Show domain picker in header</td>
<td>Places a domain picker in the banner frame, next to the user menu, when this setting is enabled.</td>
</tr>
<tr>
<td>Printer friendly version (🖨️)</td>
<td>Opens a printer friendly version of the current content frame.</td>
</tr>
<tr>
<td><strong>Theme tab</strong></td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Theme</strong></td>
<td>Select a theme for the user interface. Select the System theme to return to the default theme.</td>
</tr>
<tr>
<td><strong>Forms tab</strong></td>
<td></td>
</tr>
<tr>
<td>Tabbed forms</td>
<td>Form sections and related lists appear in tabs when this setting is enabled.</td>
</tr>
<tr>
<td>Related List Loading</td>
<td>Determines when related lists load on forms. For more information, see Configure when a related list loads on page 111.</td>
</tr>
<tr>
<td><strong>Notifications tab</strong></td>
<td></td>
</tr>
<tr>
<td>Connect Mobile Notifications</td>
<td>Users receive conversation requests on mobile devices when this setting is enabled. This setting is available if Connect on page 2261 is enabled.</td>
</tr>
<tr>
<td>Connect Desktop Notifications</td>
<td>Users receive conversation requests on desktop computers when this option is enabled. This setting is available if Connect is enabled.</td>
</tr>
<tr>
<td>Connect Email Notifications</td>
<td>Users receive an email notification of a conversation request when this setting is enabled. This setting is available if Connect is enabled.</td>
</tr>
<tr>
<td><strong>Developer tab</strong></td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>(Administrators only) Provides a link to the application picker, the Applications list, and the currently selected application.</td>
</tr>
<tr>
<td>Show application picker in header</td>
<td>Places an application picker in the banner frame, next to the user menu, when this setting is enabled.</td>
</tr>
<tr>
<td>Update Sets</td>
<td>(Administrators only) Provides a link to the Update Sets list, the Update Set picker, and the currently selected update set.</td>
</tr>
<tr>
<td>Show update set picker in header</td>
<td>Places the update set picker in the banner frame, next to the user menu, when this setting is enabled.</td>
</tr>
<tr>
<td>JavaScript Log and Field Watcher</td>
<td>(Administrators only) Opens the client-side JavaScript Debug window.</td>
</tr>
</tbody>
</table>
Figure 8: System Settings window, General tab

UI15 banner frame

The banner frame runs across the top of every page.
The banner frame for UI15 contains a logo and the following information, controls, and tools.
### Table 10: UI15 banner frame controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome message</td>
<td>Shows the name of the current user and provides a link to the user’s profile.</td>
</tr>
<tr>
<td>Impersonation key icon (👤)</td>
<td>(Administrators only) Allows the administrator to switch user views without logging out; useful for testing security and role-specific setup functions.</td>
</tr>
<tr>
<td>Elevate privileges icon (🔒)</td>
<td>(Elevated privilege roles only) Allows the user to elevate to a higher security role when needed and then return to their normal role.</td>
</tr>
<tr>
<td>Global text search icon (🔍)</td>
<td>Searches for text in multiple applications.</td>
</tr>
<tr>
<td>Collapse/Expand banner icon (🔍/🔍)</td>
<td>Alters the amount of space the banner frame occupies.</td>
</tr>
<tr>
<td>Logout</td>
<td>Returns to the Welcome page for subsequent login.</td>
</tr>
<tr>
<td>Gear icon (⚙️)</td>
<td>Displays the system menu with additional settings and controls.</td>
</tr>
</tbody>
</table>

The gear icon, located within the banner frame, displays the system menu.

### Table 11: System menu controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What's new</td>
<td>Opens a slideshow of UI15 highlights.</td>
</tr>
<tr>
<td>Help icon (❓)</td>
<td>Opens the ServiceNow Wiki in another window or tab.</td>
</tr>
<tr>
<td>Home icon (🏠)</td>
<td>Provides links to available application homepages.</td>
</tr>
<tr>
<td>Wrap Longer Text in List Columns</td>
<td>Long strings wrap in list columns instead of appearing as one long line when this setting is enabled.</td>
</tr>
<tr>
<td>Compact View</td>
<td>The UI is optimized to display more information in the browser window when this setting is enabled.</td>
</tr>
<tr>
<td>Accessibility Enabled</td>
<td>Section 508 accessibility features are activated when this setting is enabled.</td>
</tr>
<tr>
<td>Tabbed forms</td>
<td>Form sections and related lists appear in tabs when this setting is enabled. This setting previously appeared as an icon in the form header.</td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Alt/#+Click Opens List Links in Form Pane</td>
<td>Pressing Alt (PC) or Command (Mac) and clicking the record name in a list, opens the record in a new form pane tab when this setting is enabled.</td>
</tr>
<tr>
<td>Date/Time</td>
<td>The format in which dates and times appear depends on this setting. Select Calendar (for example, 2014-11-13, 15:58:58), Time Ago (for example, 11 minutes ago), or Both.</td>
</tr>
<tr>
<td>Compact list date/time</td>
<td>Date and time values appear in a compact format when this setting is enabled. The year is not shown for date values within the current year, and seconds are not shown for time values. This setting is not available if the Date/Time selection is Time Ago.</td>
</tr>
<tr>
<td>Split Layout</td>
<td>Splits the content frame into two panes, either Vertical or Horizontal. In a vertical split, the list pane is on the left and the form pane is on the right. In a horizontal split, the list pane is on the top and the form pane is on the bottom. Select None to disable the split layout.</td>
</tr>
<tr>
<td>Related List Loading</td>
<td>Determines when related lists load on forms. For more information, see Configure when a related list loads on page 111.</td>
</tr>
<tr>
<td>Application</td>
<td>(Administrators only) Provides a link to the Applications list ( ), the application picker, and the currently selected application ( ).</td>
</tr>
<tr>
<td>Update Set</td>
<td>(Administrators only) Provides a link to the Update Sets list, the Update Set picker, and the currently selected update set.</td>
</tr>
<tr>
<td>Language</td>
<td>Select a language or return to the default language ( ). This option is available if a language plugin is activated.</td>
</tr>
<tr>
<td>Time zone</td>
<td>Select a time zone or return to the default time zone ( ).</td>
</tr>
<tr>
<td>Theme</td>
<td>Select a theme for the user interface or return to the system theme ( ).</td>
</tr>
<tr>
<td>Domain</td>
<td>Specifies the currently selected domain for organizations that are using domain separation. The user must have the domain_admin role for this setting to appear.</td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Printer friendly version (狲)</td>
<td>Opens a printer friendly version of the current content frame.</td>
</tr>
<tr>
<td>JavaScript Log and Field Watcher (AndServe)</td>
<td>(Administrators only) Opens the client-side <em>JavaScript Debug window</em>. This icon is commonly called the debug icon.</td>
</tr>
</tbody>
</table>
Figure 9: System menu
## UI11 banner frame

The banner frame runs across the top of every page. It contains a logo and the following information, controls, and tools. Administrators can customize the banner frame.

![UI11 banner frame](image)

**Figure 10: UI11 banner frame**

The UI11 banner frame contains the following information, controls, and tools.

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome message</td>
<td>Shows the name of the current user and provides a link to the user’s profile.</td>
</tr>
<tr>
<td>Impersonation key icon (눈)</td>
<td>(Administrators only) Allows the administrator to switch user views without logging out. This is useful for testing security and role-specific setup functions.</td>
</tr>
<tr>
<td>Elevate privileges icon (🔒)</td>
<td>(Administrators only) Allows the user to elevate to a higher security role when needed and then return to their normal role.</td>
</tr>
<tr>
<td>Global text search icon (🔍)</td>
<td>Searches for text in multiple applications.</td>
</tr>
<tr>
<td>Collapse/expand banner icon (🔍)</td>
<td>Alters the amount of space the banner frame occupies.</td>
</tr>
<tr>
<td>Logout</td>
<td>Returns to the Welcome page for subsequent login.</td>
</tr>
<tr>
<td>Homepage icon (🏠)</td>
<td>Provides links to selectable application homepages.</td>
</tr>
<tr>
<td>Printer friendly version icon (_printer)</td>
<td>Opens a printable version of the current content frame.</td>
</tr>
<tr>
<td>Help icon (❓)</td>
<td>Opens the ServiceNow Wiki in another window or tab.</td>
</tr>
<tr>
<td>JavaScript Debugger window icon (Debugger)</td>
<td>(Administrators only) Opens the client-side JavaScript Debug window. This icon is commonly called the debug icon.</td>
</tr>
</tbody>
</table>

The **Switch UI** link may also appear in the banner next to the global text search field. This link allows switching between UI11 and the classic interface. Administrators control the appearance of the link in the banner frame with the glide.ui11.show_switch_link property. This property is set to false for new instances.
The Edge

The Edge is a toolbar on the left side of the screen which provides quick access to features such as bookmarks and flyout windows.

The Edge is available in UI15 and UI11. The collapsed view of the UI16 application navigator is similar to the Edge.

UI15 Edge buttons

In UI15, the Edge provides the following buttons.

![UI15 Edge buttons](image)

Figure 11: UI15 Edge

### Table 13: UI15 Edge buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toggle navigator</td>
<td>Shows or hides the application navigator.</td>
</tr>
<tr>
<td>List and form view</td>
<td>Splits the content frame into two vertical panes, with the list pane on the left and the form pane on the right. To split the pane horizontally, click Split Layout Horizontal.</td>
</tr>
<tr>
<td>Tagged documents</td>
<td>Displays the Tagged Documents page.</td>
</tr>
<tr>
<td>All bookmarks</td>
<td>Provides a list of all bookmarks in the Edge. Users can create, customize, and delete bookmarks. For more information, see Bookmarks on page 28.</td>
</tr>
</tbody>
</table>
### UI11 Edge buttons

In UI11, the Edge provides the following buttons.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toggle navigator</td>
<td>Shows or hides the application navigator.</td>
</tr>
<tr>
<td>Toggle banner</td>
<td>Shows or hides the banner frame.</td>
</tr>
<tr>
<td>Toggle vertical split form pane</td>
<td>Splits the content frame into two panes, with the list pane on the left and the form pane on the right.</td>
</tr>
<tr>
<td>Toggle horizontal split form pane</td>
<td>Splits the content frame into two panes, with the list pane on the top and the form pane on the bottom.</td>
</tr>
<tr>
<td>All bookmarks</td>
<td>Provides a list of all bookmarks in the Edge. Users can create, customize, and delete bookmarks. For more information, see Bookmarks on page 28.</td>
</tr>
<tr>
<td>Bookmark and pane-based UI help</td>
<td>Displays the Edge help window.</td>
</tr>
</tbody>
</table>

### Bookmarks

A bookmark is a link to information in the ServiceNow system, such as a record or a module, that is stored on the Edge.

All users create and manage their own bookmarks.

Bookmarks are available in UI15 and UI11. Favorites replace bookmarks in UI16. For more information, see Use favorites in UI16 on page 34.
Create a bookmark

You can create bookmarks by dragging links to the Edge.

Role required: none

You can create a bookmark from any of the following links:

- Modules in the application navigator
- Breadcrumbs
- Links in lists (including lists displayed in flyouts)
- Reports

Note: You may not be able to create bookmarks with other types of links. To request this functionality for another type of link, you can log an enhancement request with the ServiceNow customer support system.

Drag a link to the Edge.
The bookmark for the link appears on the Edge. Each bookmark includes an image and a title, both of which can be customized.

Customize a bookmark

You can customize bookmark appearance and behavior.

Role required: none

1. Open the bookmark settings window from any of the following locations in the interface.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bookmark tooltip</strong></td>
<td>1. Point to the bookmark in the Edge.</td>
</tr>
<tr>
<td></td>
<td>2. In the tooltip window, click Edit Bookmark.</td>
</tr>
<tr>
<td><strong>Bookmarks list</strong></td>
<td>1. Point to the all bookmarks icon in the Edge.</td>
</tr>
<tr>
<td></td>
<td>- UI15: All Bookmarks</td>
</tr>
<tr>
<td></td>
<td>- UI11: Star</td>
</tr>
<tr>
<td></td>
<td>2. Point to a bookmark in the list.</td>
</tr>
<tr>
<td></td>
<td>3. Click the edit bookmark icon by the bookmark name.</td>
</tr>
<tr>
<td><strong>Bookmark flyout</strong></td>
<td>1. Open a flyout bookmark.</td>
</tr>
<tr>
<td></td>
<td>2. Click the edit bookmark icon in the corner of the flyout.</td>
</tr>
</tbody>
</table>

2. Enter a Title.
3. To open the bookmark link in a flyout window, select the **Flyout** check box and clear the **Open in form pane** check box. By default, the bookmark opens in the content pane. Selecting **Flyout** opens a window on top of the current screen without navigating away from your working panes.

   **Note:** Homepages that are set to refresh automatically should not be added as flyout bookmarks. These homepages refresh in the background, which can slow down performance for you and other users.

4. Select the **Show on edge** check box to display the bookmark as a button on the Edge. Clear the check box to remove the button from the Edge; the bookmark still appears in the bookmark list.

5. To change the bookmark image, click **Image** and then select from the list of available images.

6. Click **Update**.

---

### Open a bookmark

When you open a bookmark, it appears in the content pane or as a flyout.

Role required: none

Open a bookmark in one of the following ways.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the bookmark as defined in the bookmark settings</td>
<td>Click the bookmark</td>
</tr>
<tr>
<td>Open the bookmark in a flyout</td>
<td>Hold Ctrl (Windows) or Command (Mac) and click the bookmark</td>
</tr>
<tr>
<td>Open the bookmark in the content pane</td>
<td>Hold Shift and click the bookmark</td>
</tr>
</tbody>
</table>

---

### Delete a bookmark

You can delete a bookmark.

Role required: none

When you delete a bookmark, it is removed from the Edge.

Delete the bookmark from either of the following locations in the interface.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmark tooltip</td>
<td>1. Point to the bookmark in the Edge.</td>
</tr>
<tr>
<td></td>
<td>2. In the tooltip window, click <strong>Edit Bookmark</strong>.</td>
</tr>
<tr>
<td></td>
<td>3. In the edit bookmark window, click <strong>Delete</strong>.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Bookmarks list</td>
<td>1. Point to the all bookmarks icon in the Edge.</td>
</tr>
<tr>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>UI15: All Bookmarks</td>
</tr>
<tr>
<td></td>
<td>1.</td>
</tr>
<tr>
<td></td>
<td>UI11: All Bookmarks</td>
</tr>
<tr>
<td></td>
<td>2. Point to a bookmark in the list.</td>
</tr>
<tr>
<td></td>
<td>3. Click the remove bookmark icon by the bookmark name.</td>
</tr>
</tbody>
</table>

### Application navigator

The application navigator, or left navigation bar, provides access to all applications and the modules they contain, enabling users to quickly find information and services.

An application is a group of modules, or pages, that provide related information and functionality in an instance. For example, the Incident application contains modules for creating and viewing incidents. The Configuration Management application contains modules for configuring servers, databases, and networks.

In UI16, the application navigator also provides access to favorites and recently viewed items.

 Administrators can customize the application navigator to provide different modules by user role, modify or define applications and modules, and change its appearance.

### UI16 application navigator

The UI16 application navigator appears at the left of the interface and provides access to all available applications and modules, favorites, and recently viewed items.

Use the application navigator to quickly find information and services.

The application navigator consists of a navigation filter and the following tabs.

#### Table 15: UI16 application navigator tabs

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All applications</td>
<td>Displays all application menus and modules. Each application appears as a section in the application navigator denoted by an application label. Modules are listed by name under each application label.</td>
</tr>
<tr>
<td>Favorites</td>
<td>Displays items you have added as favorites. If you upgrade to UI16 from a previous interface, any bookmarks you created are automatically converted to favorites.</td>
</tr>
<tr>
<td>History</td>
<td>Displays items you have recently accessed.</td>
</tr>
</tbody>
</table>
Figure 13: All applications tab

Figure 14: Favorites tab
Enter text in the navigation filter to show matching applications, modules, and favorites. Matching favorites appear at the top of the results. You can also use the navigation filter to quickly access a specific table. For more information, see *Navigate directly to a table* on page 44.

**Collapse or expand information in the UI16 application navigator**

You can collapse or expand information in the application navigator to display only what you want to see.

Role required: none

Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collapse or expand an application or application section</strong></td>
<td>Click the application or application section label.</td>
</tr>
<tr>
<td><strong>Collapse or expand all applications</strong></td>
<td>Double-click the all applications tab icon ( ).</td>
</tr>
<tr>
<td><strong>Collapse or expand the application navigator</strong></td>
<td>Click the arrow icon at the bottom of the application navigator. In the collapsed view, the application navigator displays favorites only. Click the More icon ( ) to view all favorites. Click the filter icon or the arrow icon in the collapsed view to expand the application navigator.</td>
</tr>
</tbody>
</table>
Use favorites in UI16

In UI16, you can save frequently accessed items as favorites, which are stored in the application navigator.

Role required: none

Items you add as favorites appear in the favorites tab of the application navigator, which is represented by a star icon.
Favorites also appear in the collapsed view of the application navigator as icons.

**Note:** Any bookmarks you created in a previous version of the UI are automatically converted to UI16 favorites. However, UI15 favorites are not converted to UI16 favorites and must be recreated manually.

Add a favorite in one of the following ways.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a module as a favorite</td>
<td>In the application navigator, click the star icon by a module.</td>
</tr>
<tr>
<td>Add all the modules under an application as favorites</td>
<td>In the application navigator, click the star icon by an application.</td>
</tr>
</tbody>
</table>
| Add a list as a favorite using the list context menu | 1. Open a list.  
2. Click the list context menu icon (ฎ) by the list title.  
3. Select **Create Favorite**.  
4. In the flyout, edit the name and icon as needed. |
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add a list as a favorite by dragging and dropping</strong></td>
<td>1. Open a list.&lt;br&gt;2. Drag a breadcrumb to the favorites tab of the application navigator.</td>
</tr>
<tr>
<td><strong>Add a record as a favorite using the form context menu</strong></td>
<td>1. Open a record.&lt;br&gt;2. Click the form context menu icon (≡) by the form title.&lt;br&gt;3. Select <strong>Create Favorite</strong>.&lt;br&gt;4. In the flyout, edit the name and icon as needed.</td>
</tr>
<tr>
<td><strong>Add a record as a favorite by dragging and dropping</strong></td>
<td>1. Open a record.&lt;br&gt;2. Drag the record title to the favorites tab of the application navigator.</td>
</tr>
<tr>
<td><strong>Add a different type of link as a favorite</strong></td>
<td>Drag a supported link type to the favorites tab of the application navigator. You can drag any of the following links:&lt;br&gt;• Breadcrumbs&lt;br&gt;• Links in lists&lt;br&gt;• Reports</td>
</tr>
</tbody>
</table>

You can add a different favorite for each view of a list or form.<br>The favorite is added to the favorites tab of the application navigator.

You can edit your favorites later.

**Edit favorites in the UI16 application navigator**

You can edit or delete favorites in the UI16 application navigator.<br>**Role required:** none

1. In the application navigator, click the favorites tab, which is represented by a star.<br>2. At the bottom of the application navigator, click **Edit Favorites**.<br>3. Do one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorder favorites in the list</td>
<td>Drag and drop a favorite to a new location in the list.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Customize the name or icon for a favorite | 1. Click a favorite.  
2. Customize the name and icon as needed. |
| Delete a favorite                       | 1. Click a favorite.  
2. Click Remove from favorites. |

4. Click Done or Edit Favorites.

View your navigation history

In UI16, you can view your navigation history in the application navigator.

Role required: none

Items you have accessed recently appear in the history tab of the application navigator, which is represented by a clock icon. Items appear in chronological order from most to least recently accessed.

![Figure 16: History tab](image)

History entries are stored on the Navigator History [sys_ui_navigator_history] table. The system creates history entries for many types of content, including lists, records, and homepages. Some content types are not tracked in the history, such as UI pages and other non-standard interfaces.

1. In the application navigator, click the history tab, which is represented by a clock.
2. Click an item to open it.
Configure the number of history entries displayed in the application navigator

A system property sets the maximum number of history entries displayed in the history tab of the application navigator.

Role required: admin

By default, the application navigator shows 30 history entries. You can configure the property to change this value.

1. Navigate to `sys_properties.list`.
2. Locate the `glide.ui.nav.history_length` property.
3. Edit the property `Value`.

UI15 application navigator

The application navigator appears at the left of the interface and provides access to all available applications and modules.

Each application appears as a section in the application navigator denoted by an application label. Modules are listed by name under each application label. Use the application navigator to quickly find information and services.
Figure 17: UI15 application navigator

Enter text in the navigation filter to show matching applications and modules. You can also use the navigation filter to quickly access a specific table. For more information, see Navigate directly to a table on page 44.

In the UI15 application navigator, a star appears by each module. For more information, see Use favorites in UI15 on page 40.

Collapse or expand information in the UI15 application navigator

You can collapse or expand information in the application navigator to display only what you want to see.

Role required: none

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collapse or expand an application or application section</strong></td>
<td>Click the application or application section label</td>
</tr>
<tr>
<td><strong>Collapse all applications</strong></td>
<td>Click the application menu icon and select Collapse All Applications</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Expand all applications</td>
<td>Click the application menu icon and select <strong>Expand All Applications</strong></td>
</tr>
<tr>
<td>Collapse or expand the application navigator</td>
<td>In the edge, click <strong>Toggle Navigator</strong></td>
</tr>
</tbody>
</table>

Switch perspectives in the UI15 application navigator

You can switch role perspectives to show different views of the application navigator.

A perspective filters the available applications by that role. For example, select **ITIL** to show only ITIL items such as Incidents, and filter out asset management items such as Asset Portfolio.

**Note:** Administrators can customize which perspectives are available and who can access them. For more information, see *Create or modify a menu list* on page 44.

1. In the application navigator, click the menu icon (▲).
2. Select **Roles**.
   - This option is available only if you have access to more than one role perspective.
3. Select a perspective.

Use favorites in UI15

In UI15 you can select modules as favorites to quickly filter out modules you do not use often.

Role required: none

Each module in the application navigator has a star icon to the left of the module name that can be selected (★) or deselected (☆).

**Note:** If you upgrade to UI16, you must recreate any UI15 favorites manually.

1. Click the star icons to select and deselect modules as favorites.
2. Click the filter favorites icon (★) next to the navigation filter to toggle between showing only the favorite modules and showing all modules.

By default, a module is automatically selected as a favorite when you open the module. To disable this setting, click the menu icon (▲) in the application navigator header and select **Automatically Add Favorites**.

UI11 application navigator

The application navigator appears at the left of the interface and provides access to all available applications and modules.

Each application appears as a section in the application navigator denoted by an application label. Modules are listed by name under each application label. Use the application navigator to quickly find information and services.
Enter text in the navigation filter to show matching applications and modules. You can also use the navigation filter to quickly access a specific table. For more information, see Navigate directly to a table on page 44.

In the UI11 application navigator, an icon appears by each module. Administrators can customize the icons.

Collapse or expand information in the UI11 application navigator

You can collapse or expand information in the application navigator to display only what you want to see.

Role required: none

You can collapse or expand information in the application navigator to display only what you want to see.

Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collapse or expand an application or application section</td>
<td>Click the application or application section label</td>
</tr>
<tr>
<td>Collapse all applications</td>
<td>Click the collapse all applications icon ( - )</td>
</tr>
<tr>
<td>Expand all applications</td>
<td>Click the expand all applications icon ( + )</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Collapse or expand the application navigator</td>
<td>In the Edge, click the toggle navigator icon</td>
</tr>
</tbody>
</table>

Switch perspectives in the UI11 application navigator

You can switch role perspectives to show different views of the application navigator.

A perspective filters the available applications by that role. For example, select **ITIL** to show only ITIL items such as Incidents, and filter out asset management items such as Asset Portfolio.

**Note:** Administrators can customize which perspectives are available and who can access them. For more information, see *Create or modify a menu list* on page 44.

1. In the application navigator, click the switch perspectives icon (▲). This icon is available only if you have access to more than one perspective.

2. Select a perspective.

Module icons

Modules in the UI11 application navigator have icons in addition to names. Administrators can modify the icons for existing modules and define icons for new modules.
Modify a module icon  
You can modify the icon that represents a module in UI11.

Role required: admin

**Note:** If you change the icon for a base system module, your change is preserved during software updates.

1. Navigate to **System Definition > Modules**.
2. Select the module for which you wish to modify the icon.
3. In the **Image** field, click the reference lookup icon (🔍).
4. Select the desired image file from the existing images.
5. Click **Update**.

Create a new module icon  
You create your own image and use it as a module icon in UI11.

Role required: admin

1. Use a graphics software program to create an icon file with the following specifications.

<table>
<thead>
<tr>
<th>Size</th>
<th>16 x 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>File type</td>
<td>.png,.gif,.jpg</td>
</tr>
</tbody>
</table>

2. **Upload** the new icon to the database.
3. In the application navigator, right-click the application name and select **Edit Application**.
4. **Personalize** the **Modules** related list to add the **Image** field.
5. Use the list editor to enter the file name (example, **my_image.png**).
Navigate directly to a table

You can use commands in the navigation filter to navigate directly to the list or form view of a table.

Role required: none

The commands work only for tables you are permitted to access.

1. In the navigation filter of the application navigator, enter one of the following commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;table name&gt;.list</td>
<td>Opens the list view of the table in the same window or tab.</td>
</tr>
<tr>
<td>&lt;table name&gt;.LIST</td>
<td>Opens the list view of the table in a new window or tab.</td>
</tr>
<tr>
<td>&lt;table name&gt;.form or &lt;table name&gt;.do</td>
<td>Opens the form view of the table in the same window or tab.</td>
</tr>
<tr>
<td>&lt;table name&gt;.FORM</td>
<td>Opens the form view of the table in a new window or tab.</td>
</tr>
</tbody>
</table>

For example, enter `change_request.form` to open a new change request.

2. In UI16, press the Enter key.

Create or modify a menu list

In UI15 and UI11, it is possible to switch menu lists, providing different perspectives of the applications and modules. You can create or edit these menu lists using the Menu List table.

Role required: admin

Menu lists are not used in UI16.

1. Navigate to System UI > Menu Lists.
2. Select a menu list to modify, or click New to create a new menu list.
   
   The applications that appear on the form are listed in the Applications field, one per line. Note that applications use the name of the application record, not the human-readable record that appears in the left navigation bar.

   The roles that are able to see the form are listed in the Roles field.
   
   • Specifying no roles will create a global menu list.
   • A menu list will display only the applications and modules the role has permissions to see.
   • If a role is specified to see a menu list, but role cannot view any of the applications within it, they will not see the menu list.

3. Adjust the information in the menu list as desired.
4. Click Update.

Enable or disable an application menu or module

You can enable (show) or disable (hide) an application menu or module in the application navigator.

Role required: admin

1. Navigate to System Definition > Application Menus.
2. If you are enabling an application menu, click **All** in the breadcrumbs to display both active and inactive application menus (remove the default filter condition).

3. Click the desired title. The application menu record opens and the **Modules** related list shows the modules that appear in the application navigator.

4. Enable or disable the application menu and modules as desired.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Enable or disable a specific module in the application | 1. Double-click the **Active** field beside the module name in the **Modules** related list.  
2. Set **Active** to **true** (show) or **false** (hide). |

| Enable or disable multiple modules at the same time | 1. Select the check boxes next to the module names.  
2. Select **Change active state** from the **Actions** choice list. |

| Enable or disable the entire application menu (for example, Incident or Service Catalog) | Select or clear the **Active** check box. |

| Restrict the application menu to specific roles | Use the **Roles** field. |

5. Click **Update**.

When you change application menus or modules, the application navigator automatically refreshes to display the changes.

### Define a menu category in UI15

Menu categories allow administrators to change the appearance of application menu labels. Use menu categories to make certain application menus stand out.

Role required: admin

1. **Add a property** with the following specifications.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.ui14.navigator.use_border_color</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

2. Navigate to **System Definition > Menu Categories**.

3. Click **New** or open an existing menu category.

4. In the **Style** field, define the border-color using an HTML **color name**, **hexadecimal (hex) value**, or **RGB color value**. For example, enter `border-color: blue`.

5. Optional: Enter a **Default order**.

If you create an application without specifying an order, the default order of the category is used to determine the relative position in the application navigator.

**Note:** You may need to clear the instance cache and refresh the page after saving the menu category to see CSS changes.
Define a menu category in UI11

Menu categories allow administrators to change the appearance of application menu labels with CSS styles. Use styles to make certain application menus stand out.

Role required: admin

1. Navigate to System Definition > Menu Categories.
2. Click New or open an existing menu category.
3. In the Style field, define CSS styles such as border-color, text color, and background-color. For example, enter `border-color: blue; background-color: #e3f3ff; color: white`.
4. Optional: Enter a Default order.
   If you create an application without specifying an order, the default order of the category is used to determine the relative position in the application navigator.

   **Note:** You may need to clear the instance cache and refresh the page after saving the menu category to see CSS changes.

Add an application menu to a category

There are two methods for adding an application menu to a category.

Role required: admin

   **Note:** Menu categories are deprecated in UI16. The following procedure has no effect in UI16.

Use one of the following methods to add an application menu to a menu category.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| In the menu category record, add the application menu to the Application Menus related list. | 1. Navigate to System Definition > Menu Categories.  
2. Open a menu category.  
3. In the Application Menus related list, click Edit.  
4. Use the slushbucket to add the application menu to the category.  
5. Click Save. |
| In the application menu record, enter the menu category in the Category field. | 1. Navigate to System Definition > Application Menus.  
2. Open an application menu.  
3. In the Category field, select the menu category.  
4. Click Submit. |
Create an application menu

You can create an application menu.

Role required: admin

1. Navigate to **System Definition > Application Menus**.
2. Click **New**.
3. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Defines the display name of the application menu.</td>
</tr>
<tr>
<td>Roles</td>
<td>Restricts access to the specified roles. All users can view the application menu when it is active.</td>
</tr>
<tr>
<td>Category</td>
<td>Specifies the menu category that defines the navigation menu style (default value is <strong>Custom Applications</strong>).</td>
</tr>
<tr>
<td>Hint</td>
<td>Defines the text that appears in a tooltip when a user points to this application menu.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate the application menu. Only active application menus appear in the application navigator.</td>
</tr>
<tr>
<td>Description</td>
<td>Provide a more detailed explanation of what this application does.</td>
</tr>
<tr>
<td>Other fields</td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>Defines the relative position of the application menu in the application navigator. If you do not specify an order, the default order of the menu category is used.</td>
</tr>
<tr>
<td>Default device type</td>
<td>This field is not used. You can define application menus for mobile devices in a separate table. See <strong>Define a new smartphone application menu</strong>.</td>
</tr>
</tbody>
</table>

**Note:** You may need to configure the form to see all fields.

4. Click **Submit**.
5. **Create modules** to appear in the application menu.

Only application menus that contain modules appear in the application navigator.

**Note:** You might need to configure the form to see all fields.

Create a module

You can create a module.
Role required: admin

1. Open the application menu record using one of the following methods.
   - Navigate to System Definition > Application Menus and select the application menu from the list.
   - Right-click the application label in the application navigator and select Edit Application. This is possible in UI15 and UI11 only.

2. Scroll down to the Modules related list and click New.

3. Define the module by completing the fields on the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Defines the module name. Choose a title that clearly identifies the module.</td>
</tr>
<tr>
<td>Application menu</td>
<td>Specifies the name of the application menu under which the module appears.</td>
</tr>
<tr>
<td>Image</td>
<td>Specifies an icon to appear next to the module title in the navigator. Module images are supported in UI11 only. Favorites replace module images in UI16 and UI15.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hint</td>
<td>Defines the tool tip that appears when a user points to the module name.</td>
</tr>
<tr>
<td>Visibility</td>
<td></td>
</tr>
<tr>
<td>Roles</td>
<td>Restricts module access to the specified roles. If this field is left blank, the module is visible to all users who have access to the application menu.</td>
</tr>
<tr>
<td>Active</td>
<td>Defines whether the module appears in the application navigator.</td>
</tr>
<tr>
<td>Override application menu roles</td>
<td>Allows users to access this module even if they do not have permission to view the containing application menu. Users must still meet the role requirements for this module.</td>
</tr>
<tr>
<td>Link Type</td>
<td></td>
</tr>
<tr>
<td>Link type</td>
<td>Specifies what type of link this module opens. You must specify additional information based on the link type. See Module link types.</td>
</tr>
<tr>
<td>Table</td>
<td>Specifies the table used by the module.</td>
</tr>
<tr>
<td>Order</td>
<td>Specifies the order in which the modules appear under the application.</td>
</tr>
</tbody>
</table>

**Module link types**

The **Link type** field on the Module form specifies what type of link the module opens.

**Table 17: Module link types**

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment</td>
<td>Links to the assessment-based survey you select in the Assessment reference field. See Create a survey module.</td>
</tr>
<tr>
<td>Content Page</td>
<td>Displays the content page you select in the Content page reference field. See Create a content page.</td>
</tr>
<tr>
<td>Homepage</td>
<td>Displays the homepage you select in the Homepage reference field.</td>
</tr>
<tr>
<td>Link Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTML (from Arguments)</td>
<td>Places HTML in the application navigator. This link type is used for more complicated links, where a flat URL is not customizable enough.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• The HTML (from Arguments) link type is supported in UI15 and UI11 only. In UI16, use the URL (from Arguments) link type instead.</td>
</tr>
<tr>
<td></td>
<td>• You must enter a value for the Arguments field.</td>
</tr>
<tr>
<td>List Filter</td>
<td>Displays an unpopulated list view for the table you select in the Table field. Allows users to specify a filter without loading the list first. Use the Filter field to define the default filter for the list. Use the View name field to specify a view.</td>
</tr>
<tr>
<td>List of Records</td>
<td>Displays the list view for the table you select in the Table field. Use the Filter field to define the default filter for the list. Use the View name field to specify a view.</td>
</tr>
<tr>
<td>Map Page</td>
<td>Displays the map page you select in the Map page reference field.</td>
</tr>
<tr>
<td>New Record</td>
<td>Displays a form for creating a new record in the table you select in the Table field. Use the View name field to specify a view. Use the Arguments field to apply a template. See Create a module for a template on page 733.</td>
</tr>
<tr>
<td>Run a Report</td>
<td>Runs the saved report you select in the Report field.</td>
</tr>
<tr>
<td>Script (from Arguments)</td>
<td>Runs a script, as defined in the Arguments field.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You must enter a value for the Arguments field.</td>
</tr>
<tr>
<td>Link Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Search Screen</td>
<td>Link that displays a blank form for searching records in the table. Use the <strong>View name</strong> field to specify a view.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Use the parameter &amp;sysparm_result_view=view_name to define the view the results are rendered in.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> All searches use a [starts with] query to search for matching text. Other query types are not supported in search screens.</td>
</tr>
<tr>
<td>Separator</td>
<td>Creates a division between modules. Enter a name in the <strong>Title</strong> field to add a section name that users can collapse or expand.</td>
</tr>
<tr>
<td>Single Record</td>
<td>Displays a form for a single record on the table. Use the <strong>View name</strong> field to specify a view.</td>
</tr>
<tr>
<td>Survey</td>
<td>Links to the legacy survey you select in the <strong>Survey</strong> reference field. Use the <strong>Survey overwrite</strong> check box to determine whether the survey can be taken multiple times.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The <strong>Survey</strong> link type is for use with legacy surveys only, which assessment-based surveys replace. Select the <strong>Assessment</strong> link type to link the module to an assessment-based survey.</td>
</tr>
<tr>
<td>Timeline Page</td>
<td>Displays the timeline page you select in the <strong>Timeline Page</strong> reference field. See <strong>Timeline pages</strong> on page 1024.</td>
</tr>
<tr>
<td>Link Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>URL (from Arguments)</td>
<td>Opens any URL, as defined in the <strong>Arguments</strong> field. <strong>[Optional]</strong> Use the <strong>Window name</strong> field to define a link that opens in a new window.</td>
</tr>
</tbody>
</table>

**Note:**
- For internal links, always use a relative link such as
  ./catalog_home.do?sysparm_view=catalog_default or catalog_home.do?sysparm_view=catalog_default. Do not use an absolute link to a ServiceNow instance. It creates problems when you move an update set from a development instance to a production instance because the URL still references the development instance.
- You must enter a value for the **Arguments** field.

**Additional queries**
For **List of Records** module link types, you can append additional queries to the module to further define the filter for the returned list.

For example, to filter active incidents that are assigned to the currently logged in user, use the following argument query:

```
active=true^assigned_to=javascript:gs.user_id()
```

**Example URL module that opens in a new window**
You can configure a module to open an external URL in a new browser window.

Role required: **admin**

1. Open the application menu record using one of the following methods.
   - Navigate to **System Definition > Application Menus** and select the application menu from the list.
   - Right-click the application label in the application navigator and select **Edit Application**. This is possible in UI15 and UI11 only.

2. Scroll down to the **Modules** related list and click **New**.
3. If the **Window name** field is not displayed, **configure the form** and add this field.
4. Select **URL (from Arguments)** from the **Link type** list.
5. Add the complete web address to the **Arguments** field.
6. Select an icon for the module in the **Image** field.
7. Enter _blank in the **Window name** field.
   - If this field is empty, the page opens in the content frame, which is the default behavior.
Example URL module that opens a list with a custom filter URL

To create a module that opens a list with a custom filter, sort order, and grouping, use a link type of URL (from Arguments) and create a link to a custom URL.

Role required: admin

Use the following shortcut to determine the arguments for your custom URL.

1. Navigate to the list by URL, without loading it in the standard interface  
   For example, navigate to the Incident list by entering `<base URL>/incident_list.do`.
2. Apply the desired filter, sort order, and grouping.
3. Copy the resulting URL from the browser address bar.
   For example, if you apply a filter of `active=true` with an ascending sort on `priority`, a descending sort on `opened_on`, and grouped by `assignment_group`, the address bar reads: `incident_list.do?sysparm_query=active=true^EQ^ORDERBYpriority^ORDERBYDESCopened_at^GROUPBYassignment_group`

   Note: Do not copy the base URL (the portion). Always use a relative link to prevent problems when you move an update set from a development instance to a production instance.

4. Define a module with a Link type of URL (from Arguments).
5. Paste the custom URL from step 3 into the Arguments field.
6. Click Submit.
   The module now opens the custom list in the content frame.

URL schema

Users can navigate to a record or module directly by using a URL. This topic explains the URL schema by which the system renders pages.

The basic schema for a system URL is:

The URL schema consists of the following elements.

- **base URL**: unique, secure Web address for each instance. The default format is: https://<instancename>.service-now.com.

To display a custom base URL in email notifications, you can set the glide.email.override.url property. For more information, see [Additional email properties](#) on page 2737.

- **nav_to.do?uri=** (optional): loads the page in the standard interface, with the banner frame on top and the application navigator on the left.
- **<page name>.do?:** where the page name is the form, list, UI page, or other page to open. To view a list, use <table name>_list.do.
- **sys_id=<sys_id>** (optional): where <sys_id> is the sys_id of the record to open in form view. To create a new record, specify a sys_id of -1.
- **<page parameters>** (optional): may specify a query, view, redirection page, and more. Sample parameters include the following.
  - `sysparm_view=ess`: specifies a view (ess).
  - `sysparm_query=number=INC00040`: specifies a query (number is INC00040).
  - `sysparm_query=priority=2^active=true`: specifies a complex query with two terms (priority is 2 and active is true).
  - `sysparm_query=priority=2^active=true^EQ^GROUPBYcategory`: groups query results (by category).
  - `sysparm_order=number`: specifies the field by which to sort (number).
  - `sysparm_order_direction=desc`: specifies a sort order (descending).
  - `sysparm_force_row_count=5`: limits the maximum number of results (5 records).
  - `sysparm_result_view=viewname`: specifies the view for search results.

- `%26CSV`: specifies a file format, can be CSV, XML, PDF, or UNL (Unload).
- `&`: separates page parameters.
- `^` (carat): builds multiple term queries or specifies multiple field values.

### Forms vs. lists (sysparm_query)

The sysparm_query page parameter behaves differently for a list versus a form.

- **List**: returns records that match the query conditions.
- **Form** (with sys_id=-1 specified): applies the values to the new record.

**Note**: If you are using the default table templates where the template name matches the table name, such as change_request or incident, using sysparm_query for that form does not work.

### Enable tiny URL support

The default URLs by which the system renders pages may exceed the character limit of Microsoft Internet Explorer, resulting in an error message. To prevent this error, you can enable tiny URL support, which generates shortened internal URLs.

**Role required**: admin

The Tiny URL Support plugin is activated automatically but is not enabled. Activate this plugin if Microsoft Internet Explorer displays failure to open page errors during routine operations in the ServiceNow platform.
Note: The system does not convert all URLs to tiny URLs. Only some URLs the system generates as redirects are converted. For example, a URL the browser generates when a user opens a record is not converted to a tiny URL.

1. Navigate to **System Properties > System**.
2. Select the **Use tiny URLs when a redirect URL becomes too large.** This ensures that URLs that are too large for IE (greater than 2083) are not used. Instead, they are converted to a tiny URL to work around the IE issue. (glide.use_tiny_urls property).
3. Optional: Update the value in the **Minimum length of a redirect URL that is turned into a tiny URL (default=1024)** (glide.tiny_url_min_length property), if desired.
4. Click **Save**.

**Examples of navigating by URL**

A list of example URLs that demonstrate ways to open pages in the ServiceNow application.

**Open a form with preset values**

**Schema:** https://<baseURL>/nav_to.do?uri=<table name>.do?
sys_id=-1%26sysparm_query=<field=value>

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opens a new Incident form in the standard interface with a priority of 1 and an incident state of <em>Awaiting Problem</em>.</td>
<td>https://&lt;instance name&gt;.service-now.com/nav_to.do?uri=incident.do?sys_id=-1%26sysparm_query=priority=1^incident_state=3</td>
</tr>
<tr>
<td>You can also use JavaScript to access GlideSystem methods. The following example creates the same type of incident as above, and also populates the caller ID with the current user ID.</td>
<td>https://&lt;instance name&gt;.service-now.com/nav_to.do?uri=incident.do?sys_id=-1%26sysparm_query=priority=1^incident_state=3^caller_id=javascript:gs.getUserID()</td>
</tr>
</tbody>
</table>

**View a list of incidents**

Schemas for the following examples:

- https://<baseURL>/nav_to.do?uri=<table name>_list.do
- https://<baseURL>/<table name>_list.do

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opens a list of all incidents with (example 1) the navigation frame.</td>
<td>https://&lt;instance name&gt;.service-now.com/nav_to.do?uri=incident_list.do</td>
</tr>
<tr>
<td>Opens a list of all incidents without the navigation frame.</td>
<td>https://&lt;instance name&gt;.service-now.com/incident_list.do</td>
</tr>
</tbody>
</table>
View a list of attachments

Schema: https://<baseURL>/nav_to.do?uri=<table name>_list.do

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opens the Attachments table in list view.</td>
<td>https://&lt;instance name&gt;.service-now.com/sys_attachment_list.do</td>
</tr>
</tbody>
</table>

View a filtered list

Schema: https://<baseURL>/nav_to.do?uri=<table name>_list.do?sysparm_query=<field=value>

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a list of active incidents with high escalation in the standard interface.</td>
<td>https://&lt;instance name&gt;.service-now.com/nav_to.do?uri=incident_list.do?sysparm_query=active=true^escalation=2</td>
</tr>
</tbody>
</table>

Return a file

Schema: https://<baseURL>/nav_to.do?uri=<table name>_list.do?sysparm_<sysparmTypeOrField=value>%26CSV

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a comma-separated value file of records in the Incident table that meet the query conditions.</td>
<td>https://&lt;instance name&gt;.service-now.com/nav_to.do?uri=incident_list.do?sysparm_query=active=false%26CSV</td>
</tr>
</tbody>
</table>

Lists

A list displays a set of records from a table. Users can search, sort, filter, and edit data in lists. Lists can be embedded in forms and may be hierarchical (have sublists).

The list interface consists of a title bar, filters and breadcrumbs, and columns of data. Each column in a list corresponds to a field on the table.

A response time indicator (in UI16 and UI15, in UI11) appears at the bottom right of some lists to indicate the time that it took to display the list.
### Figure 20: Record list

#### List features and actions

The list interface consists of a title bar, filters and breadcrumbs, and columns of data. Each of these components provides features and lets you take action on the list and the displayed records.
Figure 21: List view features, menus, and actions

The following topics provide more information about list components and the features and actions you can perform.

List title bar

The title bar displays the name of the list and provides access to several controls.

List title menu

You can click the title of the list or the menu icon ( ) to the left to access options related to viewing and filtering the list. Administrators can customise the options that appear in the list title menu.
Table 18: List title menu options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Changes the view of the list by presenting different information.</td>
</tr>
<tr>
<td>Filters</td>
<td>Changes the values in the filter. Some of the choices are: None, Active, and Edit personal filters.</td>
</tr>
<tr>
<td>Group by</td>
<td>Groups records in a list by the values in a selected field from that table. Any field from the table can be used as a group filter, whether or not it appears in the list.</td>
</tr>
<tr>
<td>Show</td>
<td>Changes the number of rows shown on each page of the list.</td>
</tr>
<tr>
<td>Refresh List</td>
<td>Refreshes the list to show changes immediately.</td>
</tr>
<tr>
<td>Create Favorite</td>
<td>Adds a link to this list to your favorites.</td>
</tr>
</tbody>
</table>

Other list title bar controls

- **New** button: Opens a blank form that allows users to create a record.

  Clicking the New button in a filtered list automatically applies the same filter to the new record. For example, in a list filtered for active, priority 1 incidents, clicking New opens a new record preset with Active selected and Priority set to 1 - Critical. You can change the preset values. If you do not want to populate a field in this way, add the following dictionary attribute to the field: ignore_filter_on_new=true.

- **Go to** (UI16 and UI15) or **Search**: Finds information in the current list.
List activity stream (✓): Shows recent record activity for all records displayed on a list.

Column headings

Column headings appear at the top of each column. These headings display column names and provide some list controls.

Column headings are stationary at the top of the list and do not scroll with list content in UI16 or UI15.

Note: Some of the options displayed on the list context menu depend on the user role and the installed applications.

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorting</td>
<td>Click the column name to sort the list in ascending order. Click again to sort in descending order. An arrow by the column name indicates the column currently being sorted as well as the sort direction. A downward pointing arrow indicates that the column is sorted in descending order.</td>
</tr>
<tr>
<td>Title</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>List context menu</td>
<td>Access this menu, also called a right-click menu, by clicking the menu icon (≡) at the top of a list column (UI16/UI15) or by right-clicking the column heading (UI11). The list context menu offers the following controls.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Sort (a to z) and (z to a)</strong>: Sort in ascending and descending order. <strong>Show Visual Task Board</strong>: Create a visual task board based on the current list. <strong>Group By</strong>: Aggregate records by a field. For more information, see <em>Use a grouped list</em> on page 89. <strong>Bar Chart</strong> and <strong>Pie Chart</strong>: Create quick bar and pie chart reports based on the filter criteria of the list. Users can then modify these reports or create gauges (depending on access rights). <strong>Configure</strong> (requires access rights): Provides administrative functions related to the information displayed and how it is controlled. <strong>Import</strong> (administrators only): Import data from an Excel template file. <strong>Export</strong>: Exports data to Excel, CSV, XML (administrators only), or PDF. For more information, see <em>Export data</em> on page 1494. <strong>Update Selected</strong> and <strong>Update All</strong> (administrators only): Change applicable field values. <strong>Create Application Files</strong> (administrators only): Creates demo data from the current list of records that can be included when you install or update the application on another instance. Used with custom application development. <strong>Import XML</strong> (administrators only): Imports a set of records that were exported from a source instance, for example, a developer instance. For more information, see <em>Import a single XML file containing one or more records</em> on page 1495.</td>
</tr>
<tr>
<td>Personalize List (燔)</td>
<td>Customizes the list layout for the current user. For more information, see <em>Personal lists</em> on page 93.</td>
</tr>
<tr>
<td>Search ( in UI16/UI15)</td>
<td>Enables the column search in UI16 and UI15. For more information, see <em>Search by one or more columns in a list</em> on page 86.</td>
</tr>
</tbody>
</table>
List fields

Fields display data and provide certain functions.

**Note:** Some of the options displayed on the field context menu depend on the user role and the installed applications.

### Table 20: Field functions

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Links</td>
<td>Open the associated record in form view. Click the reference icon or first column field to open the current record. You can also click a link to a related record in other columns to go to that related record.</td>
</tr>
<tr>
<td>Editing</td>
<td>Changes the information in one or more records. For more information, see <em>Methods for list edits</em> on page 90.</td>
</tr>
<tr>
<td>Reference icon ( in UI16 and UI15, in UI11)</td>
<td>Provides detailed information about the record. Point to the reference icon to open a pop-up window, or click it to open the record in form view. For more information, see <em>Reference Icon.</em></td>
</tr>
</tbody>
</table>
| Context menu                  | Also called a right-click menu, offers the following options:  
  - **Show Matching** and **Filter Out:** Provide *quick filter* options.  
  - **Copy URL to Clipboard:** Copies the URL for the form view of the record to the clipboard. Follow browser instructions if browser security measures restrict this function.  
  - **Copy sys_id** (administrators only): Copies the sys_id of the record to the clipboard. Follow browser instructions if browser security measures restrict this function.  
  - **Assign Tag** (UI16 and UI15): Allows the user to assign a new or existing tag or label to a record, which provides quick access to frequently referenced or urgent information. When a tag is assigned to a record in UI16 or UI15, the record is displayed on the Tagged Documents page. In UI11, a link to the record is displayed in the application navigator under the assigned tag.  
  - **Assign to me, Approve,** and **Reject:** Provide *quick edit* options.  
  - **Add to Visual Task Board:** Allows users to add the selected record to *Visual Task Boards* they own. |
**Detail rows**

When detail rows are enabled, they appear below the field row for each record and display the value of a specified field. For example, the detail row can display the short description for each incident in a list.

Detail rows support the same functionality as fields, including links, editing capabilities, and access to the context menu.

![Figure 23: Detail rows](image)

Administrators can enable detail rows and add them to lists (UI15 or later required). For more information, see *Administer detail rows* on page 688.

---

**Note:** When a field is designated as the source for the list detail rows, the system hides the list column for that field.

**Configure list field status indicators**

Field status indicators are used to highlight certain fields on lists and to provide status information.

Role required: none

- In UI16 and UI15, field status indicators on lists are displayed with modern cell coloring, as a colored circle on the left side of the field. You can revert to using a field background color by personalizing the list and disabling modern cell coloring.

![Figure 24: Modern cell coloring on](image)

- In UI11, field status indicators on lists are displayed using a field background color.

![Figure 25: Modern cell coloring off](image)

All UI versions display field status indicators for mandatory fields only.
You can select the style of the field status indicator in UI16 or UI15.

1. Click the personalize list icon (でしょうか) in the upper left corner of a list.
2. Select the **Modern cell coloring** check box to use the UI16/UI15 style field status indicator. Clear this check box to use the UI11 style.
3. Click **OK**.

For more information, see *Personal lists* on page 93.

**Pop-up forms**

Pop-up forms enable you to edit records without leaving the list in UI16 or UI15.

To display a pop-up form, press the Shift key and click a field link or point to the reference icon for a record. You can edit the pop-up form like any other form.

![Pop-up form for editing records](image-url)

**Figure 26: Pop-up form for editing records**
Action check boxes

Action check boxes enable you to perform actions on selected items in a list.
Role required: none

Figure 27: Action check boxes and action choice list

1. Select the check boxes beside the records you want to affect. To select all records on the page, select the check box beside the action choice list at the bottom of the list.
2. Apply the desired action.
   • In the column context menu, select Update Selected to update all the selected records.
   • UI11: in the column context menu, Update Selected records using an editing form.
   • In the action choice list, select an action such as Delete (administrators only), Show on Live Feed, or Assign Tag. The available actions vary depending on the list and which plugins are activated.

Hierarchical lists

Hierarchical lists allow users to view records from related lists directly from a list of records without navigating to a form.

Lists can have sublists in a hierarchy that can also be accessed in list view. To expand or collapse the related lists on a record in a hierarchical list, click the arrow (_above) beside the reference icon.
Use embedded lists

Some lists are embedded in forms.

Role required: none

Changes to embedded lists are saved when the form is saved. For more information, see *Edit a form* on page 121.

Use the following controls to work with an embedded list.

- To expand or collapse an embedded list, click the expand (.expand) or collapse (collapse) icon in the list header.
- To add a row, double-click **Insert a new row**.
- To edit a row, double-click in an empty area of the field. See *Use the list editor* on page 90.
- To delete a row, click the delete icon (delete) beside the row.

New rows are removed immediately and existing rows are designated for deletion when the record is saved. To clear this designation, click the delete icon again.
Customize the number of list rows per page

You can customize the number of records, or rows, to display on each page of a list. The default is 20 rows per page.

Role required: none

When you customize the number of rows, it applies to all lists that you can access.

For information about the performance impact of displaying more rows, see the blog post *Improve performance by displaying “just enough” data* by a ServiceNow employee in the ServiceNow Community.

[edit]

1. Open a list.
2. Click the list title menu and select **Show**, and then select the number of rows to show on a page. The list refreshes to display the number of records selected, or the total number of records if there are fewer than the number of rows you are displaying. The page control is updated to show your selected number of records (1 to 50 of 59).

**Activity streams in list view**

You can stream live activity information for all records on the current list in UI16 or UI15.
**Figure 30: Activity stream window**

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>State</th>
<th>Approval</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHG0000001</td>
<td>Rollback Oracle Version</td>
<td>Requested</td>
<td></td>
</tr>
<tr>
<td>CHG0000004</td>
<td>Upgrade to Oracle 11i</td>
<td>Not Yet Requested</td>
<td></td>
</tr>
<tr>
<td>CHG0000005</td>
<td>Install new PBX</td>
<td>Approved</td>
<td></td>
</tr>
<tr>
<td>CHG0000006</td>
<td>Put another 100 Gb drive on the 2nd Floor Server</td>
<td>Not Yet Requested</td>
<td></td>
</tr>
<tr>
<td>CHG0000007</td>
<td>R&amp;D wants to know what it’d cost to switch them over to Linux desktops</td>
<td>Rejected</td>
<td></td>
</tr>
<tr>
<td>CHG0000008</td>
<td>Install new Cisco</td>
<td>Requested</td>
<td></td>
</tr>
<tr>
<td>CHG0000009</td>
<td>Apply patches 10.2.0.1 to 10.2.0.3</td>
<td>Approved</td>
<td></td>
</tr>
<tr>
<td>CHG0000010</td>
<td>Java Application Server change</td>
<td>Approved</td>
<td></td>
</tr>
<tr>
<td>CHG0000011</td>
<td>Another Java Application Server change</td>
<td>Approved</td>
<td></td>
</tr>
</tbody>
</table>
To view this information, click the list activity stream icon in the list title bar. This icon appears in the title bar for all task tables.

The activity stream information appears in a flyout window, and is the same information that appears in the activity formatter for a record. The information in the flyout window updates automatically with audit and journal entries. Click the >> icon at the top to close the activity stream.

You can add a comment to any item in the activity stream. When you point to the item with your cursor, a Comment button appears.

Switch between list views

A view defines the elements that appear when a user opens a list.

Role required: none

You can switch between list views to which you have access. For example, you can select the Mobile view where it is available and adjust the size of the browser window to see how it looks on a mobile device.

1. Open the list.
2. Right-click the form header and select View, and then select the view from the list.
   The page refreshes with the selected view.

Filters and breadcrumbs

A filter is a set of conditions applied to a table to help you find and work with a subset of the data in that table.

You can apply, modify, create, and save filters. A hierarchical list of conditions at the top of the table—breadcrumbs—indicates the current filter.

Breadcrumbs offer a quick form of filter navigation. The left condition is the most general, and the right condition is the most specific. Clicking a breadcrumb removes all the conditions to its right. Clicking the condition separator (>) before a condition removes only that condition.

![Breadcrumbs](image)

**Figure 31: Breadcrumbs**

- Clicking **Priority = 2** removes the condition **Category = Software** and returns all active incidents with a priority of 2.
- Clicking the condition separator (>) before **Priority = 2** removes the condition **Priority = 2** and returns all active incidents in the software category.
• Clicking **All** removes all conditions and returns all incidents in the system.

Click a breadcrumb to refresh the list of records and show the latest information from the database for those records.

---

**Note:** Clicking the **New** button in a filtered list automatically applies the same filter to the new record. For example, in a list filtered for active, priority 1 incidents, clicking **New** opens a new record preset with **Active** selected and **Priority** set to **1 - Critical**. You can change the preset values on the form as needed. If there is a field you do not want to have populated in this way, you can add the following dictionary attribute to the field: `ignore_filter_on_new=true`.

Additional navigational functions are available when you right-click a breadcrumb.

### Table 21: Breadcrumb right-click options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open new window</td>
<td>Opens the results list for the breadcrumb in a new tab or window.</td>
</tr>
<tr>
<td>Copy URL</td>
<td>Copies the URL to the clipboard for the results list of the breadcrumb. Follow browser instructions if browser security restricts this function.</td>
</tr>
<tr>
<td>Copy query</td>
<td>Copies the encoded query to the clipboard for the breadcrumb. You can use this query in the URL of an instance or in the reference qualifier field of a dictionary entry. For example, if you are viewing a list of all active incidents with a high or medium impact, right-click the breadcrumb and copy the query <code>active=true^impact=1^ORimpact=2</code>. You can append this query to the end of the instance URL as follows: <code>https://{instance_name}/incident_list.do?sysparm_query=active=true^impact=1^ORimpact=2</code> This selection is not available for the <strong>All</strong> breadcrumb.</td>
</tr>
</tbody>
</table>

---

**Quick filters**

To quickly filter a list using a value in a field, right-click in the field and select **Show Matching** or **Filter Out**. For date fields, choose from **Show Before**, **Show After**, and **Filter Out**.

These functions add a condition to the right in the breadcrumb of the current filter.
In this example, right-clicking Active and selecting Show Matching adds the condition Incident state = Active as the most specific condition of the filter. By contrast, right-clicking Active and selecting Filter Out adds the condition Incident state != Active as the most specific condition of the filter.

For date and date-time fields you can also use Show After or Show Before to define a time-based filter.

Create filters

A filter restricts what records appear in a list by providing a set of conditions each record must meet to be included in the list.

Role required: none

A condition consists of the following parts.

- **Field**: Each field contains data from a particular column in the table. Selecting a reference field allows you to dot-walk to data from other tables.
- **Operator**: Each field type has its own set of valid operators. The operator determines if a value is needed.
- **Value**: Each field has its own set of valid values determined by the field type. Reference fields have access to auto-complete, and choice lists provide a list of options.
- **Grouping**: Each condition line is grouped with either an AND or OR connector. The filter requires all condition lines linked with an AND connector to be met. The filter separately evaluates each condition line linked with an OR connector.

Create filters on a list using the condition builder. To make the condition builder appear every time you open the list, click the pin/unpin filter icon ( in UI16 and UI15, and  in UI11).

1. Open the condition builder in one of the following ways.
   - Click the show / hide filter icon ( ) beside the breadcrumbs in UI16 or UI15.
   - Click the arrow ( ) beside the breadcrumbs in UI11.
2. Select a field from the list.
The field type determines the available operators and values. For example, the **Active** field can have a value of **true**, **false**, or **empty**, while a text field can have many different values. Similarly, the **greater than** operator does not apply to the **Active** field, but it does apply to the **Priority** field. For more information, see *Condition builder* on page 157.

3. Select an **operator** from the list.
4. Select or enter a **value**, if appropriate.
5. Add or remove conditions to construct the desired filter by completing one or more of the following steps.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To add a top-level condition</td>
<td>Click <strong>AND</strong> or <strong>OR</strong> on the condition builder toolbar above the conditions.</td>
</tr>
<tr>
<td>To add a dependent condition</td>
<td>Click <strong>AND</strong> or <strong>OR</strong> beside the condition.</td>
</tr>
<tr>
<td>To remove a condition</td>
<td>Click the delete icon (X) beside the condition.</td>
</tr>
</tbody>
</table>

6. To specify the sort order of the results, click **Add Sort**, and then select a field to sort by and a sort order.
7. Optional: Click **Save** to keep the filter for future use.
8. Click **Run** to apply the filter.

**Note:** To find all records that do not contain the specified value, create a filter with two conditions: `[field] [is not] [value] OR [field] [is] [empty].`

**OR conditions**
The condition builder uses two different types of OR conditions, top-level and dependent.

Using a dependent OR condition, you can specify alternative criteria to a single operation. Dependent OR conditions work in the manner A and (B or C).

For example, to return a list of all unassigned problem and incident records from the Task table, create a filter with a dependent OR on the **Number** field.

- [Assigned to] [is] [empty] AND [Number] [begins with] [PRB] OR [Number] [begins with] [INC].
Figure 33: Dependent OR

A top-level OR condition allows you to display the results of multiple filter criteria in a single list. Top level OR conditions work in the manner (A and B) or (C and D).

For example, to return a single list of all active incidents with a category of hardware, and all inactive incidents with a category of software, create two condition sets separated by a top-level OR condition.

- [Active] [is] [true] AND [Category] [is] [Hardware]
- Top level OR condition
- [Active] [is] [false] AND [Category] [is] [Software]
Figure 34: Top level OR

Top-level and dependent OR conditions can be used together. Filters using both types of OR conditions work in the manner (A or B) or (C or D). By mixing AND conditions with top-level and dependent OR conditions, you can create very specific filters.

Filter on multiple string values
For a string field, you can create a filter that searches for multiple values by creating a comma-delimited list.

This feature enables administrators to copy and paste search criteria from a Microsoft Excel spreadsheet into a filter, for example.

**Note:** Do not use the `[is one of]` operator on fields that contain commas, as the query does not return the expected set of records. Instead, create a filter using multiple `[or]` statements.

1. Create the filter with the `[is one of]` or `[is not one of]` operator.
   Depending on the selected field, a choice list or a text box appear.
2. Select one or more of the options by using multiple selection key commands.
   The choice list remains visible.
Alternatively, for text or number fields, type your search options. Separate the options by commas or put each option on a separate line, and do not enclose the selections in brackets.

3. Click **Run** to filter the list.
The filter conditions appear as a comma-delimited string at the top of the results list.
Figure 35: Comma-delimited filter string

Dynamic operators
The dynamic operator, **is (dynamic)**, lists predefined dynamic filter options where the condition value is computed from a value in a reference field.

The following dynamic filter options are available by default.

**Note**: Administrators can create new dynamic filter options.

<table>
<thead>
<tr>
<th>Target table of reference field</th>
<th>Option label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User [sys_user]</td>
<td>Me</td>
<td>The reference field contains the current user.</td>
</tr>
<tr>
<td></td>
<td>One of My Assignments</td>
<td>The reference field contains the current user or someone for whom the current user is a delegate for assignments.</td>
</tr>
<tr>
<td></td>
<td>One of My Approvals</td>
<td>The reference field contains the current user or someone for whom the current user is a delegate for approvals.</td>
</tr>
<tr>
<td>Target table of reference field</td>
<td>Option label</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>Users With Roles</td>
<td>Users With Roles</td>
<td>The reference field contains users that have any role.</td>
</tr>
<tr>
<td>Group [sys_user_group]</td>
<td>One of My Groups</td>
<td>The reference field contains a group to which the current user belongs.</td>
</tr>
</tbody>
</table>

The following table contains examples and descriptions of dynamic filter conditions.

**Table 23: Example of using the dynamic operator**

<table>
<thead>
<tr>
<th>Field</th>
<th>Operator</th>
<th>Dynamic filter option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caller</td>
<td>is (dynamic)</td>
<td>Me</td>
<td>Computes the value of Caller based on the current user viewing the list.</td>
</tr>
<tr>
<td>Assignment group</td>
<td>is (dynamic)</td>
<td>One of My Groups</td>
<td>Computes the value of Assignment group based on the current user viewing the list.</td>
</tr>
</tbody>
</table>

Save and use filters in a list view

Depending on your access rights, you may save filters for everyone, a user group, or yourself. You can apply saved filters and edit or delete them.

Role required: varies depending on action. See procedure.

1. Create or modify a filter in the filter interface.
2. Click **Save**.
3. Enter a name for the filter.
4. Select one of the following visibility options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Me</td>
<td>Creates a personal filter, which only you can access. This option is available to all users.</td>
</tr>
<tr>
<td>Everyone</td>
<td>Creates a global filter, which all users can access. This option is available to users with the filter_global role.</td>
</tr>
<tr>
<td>Group</td>
<td>Creates a group filter, which only members of the user group you select can access. This option is available to users with the filter_group role.</td>
</tr>
</tbody>
</table>

5. Click **Save**.
6. To use a saved filter, select the filter name in the list title menu. The filter runs and the breadcrumbs appear.
7. To edit a filter, select **Edit personal filters** from the list title menu and complete the following steps.
   a) Click the title of the filter to edit.
   b) Update the filter title or add or remove conditions.
   c) Optional: To delete the filter, click **Delete** and confirm deletion.
   d) If you edited the filter, click **Update**.

**Filter admin functions**

Administrators can work with filters in the Filters table [sys_filter], including creating, editing, and deleting filters.

Administrators can also create scripted filters that cannot be created in the condition builder. Scripted filters required a knowledge of JavaScript. Scripted filters can be used to create additional *dynamic filter options*.

*Create and edit filters*

You can create a filter and edit any saved filter.

Role required: admin

1. Navigate to **System Definition > Filters**.
2. To create a filter, complete the following steps.
   a) Enter a **Title** and select the **Table**.
   b) Add filter conditions.
   c) Click **Submit**.
3. To edit a saved filter, select the filter to edit and complete the following steps.
   a) Modify the **Filter** conditions as necessary.
   b) Click **Update**.
Delete filters
You can delete any saved filter, including global, group, or personal filters.

Role required: admin

1. Navigate to System Definition > Filters.
2. Optional: To see who created the filter and when, personalize the list to add the created_by and created fields to the list.
3. Optional: To see who has access to the filter, configure the list to add the user, group, or domain fields.
   Filters that are not assigned to a user or group are global.
4. Select the filter to delete.
5. Click Delete and confirm the deletion.

Create scripted filters
The condition builder alone cannot create some filters, such as displaying a record set that is dependent on an unrelated table. You can create JavaScript functions for use in advanced filters if you have a knowledge of JavaScript.

Role required: admin

1. Create a new script include.
2. In the Script field of the script include, create a JavaScript function that returns an array of sys_ids.
   • Ensure that the function uses the same name as the script include.
   • Ensure that the script include is Active and Client callable.
3. Call the JavaScript function from the condition builder.
   For more information, see GlideRecord on page 3960 queries and Script includes on page 3862.

A company provides intensive care for a group of customers. To track these services, the service manager needs a high-level journal and links to all incidents that the customers raise.

The company creates a new application, Intensive Care, and a table, [u_intensive_care]. While the table contains a reference field for the customer name, there is no direct link to the user table. Thus, the manager cannot set up an incident list filter using the condition builder for customers who are under intensive care.

The solution is to write a JavaScript function that uses a GlideRecord query to build an array of user sys_ids in the [u_intensive_care] table, as shown in the sample code below. Call the function from the condition builder in the Incident table ([Caller] [is] [javascript:myFunction()]).

```javascript
function myFunction() {
  var arrUsers = [ ];
  gr = new GlideRecord ( 'u_intensive_care' );
  gr. query ( );
  while (gr. next ( ) ) {
    arrUsers. push (gr. u_customer. toString ( ) );
  }
  return arrUsers;
}
```

Create a dynamic filter option
Dynamic filter options enhance filters by allowing users to run a set query against a reference field without having to enter JavaScript code in the condition builder.
Role required: admin

All dynamic filter options use the **is (dynamic)** operator and call a particular scripted filter.

**Note:** Dynamic filters are not supported in scoped applications.

---

**Figure 37: Dynamic operands**

For more information about using dynamic operators, see *Dynamic operators* on page 77.

1. Create a scripted filter as a client-callable script include or business rule.
2. Navigate to **System Definition > Dynamic Filter Options**.
3. Click **New**.
4. Complete the form.

**Table 24: Dynamic filter options**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Enter the text you want to appear as an option when a user selects the <strong>is (dynamic)</strong> operator.</td>
</tr>
<tr>
<td>Script</td>
<td>Enter the name of the function you created.</td>
</tr>
<tr>
<td>Field type</td>
<td>Select <strong>Reference</strong>.</td>
</tr>
<tr>
<td>Referenced table</td>
<td>Select the table this filter option applies to.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to designate the placement of this option in the filter option choice list.</td>
</tr>
<tr>
<td>Roles</td>
<td>Select the role a user must have to see this option.</td>
</tr>
<tr>
<td>Active</td>
<td>Enable or disable the option.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Reference script</td>
<td>Select the client-callable script include or business rule you created for the scripted filter.</td>
</tr>
<tr>
<td>Available for filter</td>
<td>Select this option to display the option as a filter breadcrumb.</td>
</tr>
<tr>
<td>Available for default</td>
<td>Select this option to allow it to be a default in a dictionary entry.</td>
</tr>
<tr>
<td>Available for ref qual</td>
<td>Select this option to allow it to be selected as a reference qualifier in a dictionary entry.</td>
</tr>
</tbody>
</table>

5. Click **Submit**.

**Encoded query strings**

An encoded query string represents a complex filter on a list of records.

Use encoded query strings to include a filter as part of a URL parameter, such as the `sysparm_query` URL parameter, or as a reference qualifier to restrict the data that is selectable for a reference field.

You can create encoded query strings manually or copy them from list filters.

**Table 25: Encoded query string examples**

<table>
<thead>
<tr>
<th>Title</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referring to Boolean or string values</td>
<td>To create a query string on Boolean fields, use this syntax: field=true or field=false. For example, to return only active records in a table, use:</td>
</tr>
<tr>
<td></td>
<td>active=true</td>
</tr>
<tr>
<td></td>
<td>To create a query string with a field that has a specific string value, use this syntax: field=value. For example, to return incident records where the category is network, use:</td>
</tr>
<tr>
<td></td>
<td>category=network</td>
</tr>
<tr>
<td>Referring to reference fields</td>
<td>To create a query string that refers to a reference field in another table, use this syntax: field_in_referenced_table=value. For example, to return users with the itil role in the <strong>Assigned to</strong> reference field on the Incident form, use:</td>
</tr>
<tr>
<td></td>
<td>roles=itil</td>
</tr>
<tr>
<td>Title</td>
<td>Example</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Referring to choice list values           | To create a query string that refers to a choice list option, use the value of the choice list, not the label, with this syntax: choice list field=value. For example, to return configuration items (CIs) with the status **Installed**, which has a value of 1 in the choice list, use:  

`install_status=1`

See *Values to associate with choice labels for scripting* on page 848 to find the options for the values. |
| Using multiple conditions                 | To use two conditions joined together by an AND operator, use the carat ^ symbol in this syntax: field1=value1^field2=value2. For example, to return all active users with the "itil" role, use:  

`active=true^roles=itil`

To use two conditions joined together by an OR operator, use the carat symbol with OR with this syntax: field1=value1^ORfield1=value2. For example, to return all users with either the "itil" or "admin" roles, use:  

`roles=itil^ORroles=admin`

To add multiple conditions where one condition is a JavaScript function, use this syntax: javascript:'field=value^' + function(). The function must return a string that can be concatenated with the first string to produce a valid query string. For example, to return all active users and meet the conditions of the getGroupQualifier() function, use:  

`javascript:'active=true'' + getGroupQualifier()` |
<table>
<thead>
<tr>
<th>Title</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listing values in order</td>
<td>To sort data by a specific field, use the ORDERBY condition with the following syntax: field1=value^ORDERBYfield2. For example, to return the days of the week in order from the Day of the Week [sys_cal_unit] table, use: unit_name=day^ORDERBYvalue In the Day of the Week table, unit_name is the field that specifies day, week, or month, and value is a numerical value for each day from 1 (Monday) to 7 (Sunday). The query string returns a list of the days of the week in order from the lowest value, which represents Monday, to the largest value, which represents Sunday.</td>
</tr>
</tbody>
</table>

**Note:** Reference qualifiers do not support the ORDERBY condition. In reference qualifiers, you can sort the reference lookup list by using standard list controls. To specify the order of an auto-complete list for a reference field, use the ref_ac_order_by dictionary attribute.

Generate an encoded query string through a filter

You can generate an encoded query string through a filter on any list and paste the string into a URL query or a reference qualifier.

Role required: none

1. Open a list of records.
2. **Construct the filter.**
3. Click **Run**.
4. Right-click the end of the filter breadcrumb and select **Copy query** from the context menu.
5. Copy the query to your system clipboard.
6. Use the query string to construct a URL or an advanced reference qualifier.
   When you use the CONTAINS operator on a list filter, the system translates the filter to a LIKE query. For example, if you filter for active records with numbers that contain 123, the URL is https://InstanceName.service-now.com/incident_list.do?sysparm_query=active%3Dtrue%5EGOTOnumberLIKE123.

Search a list

You can search a list to find information quickly.

Role required: none

The list title bar includes options for searching the list. Administrators can enable text searches for any list.

There are two options for searching the list.

- The Go to option sorts the list by the selected field and returns records where the field value is equal to or greater than the search term. For the Number field, which is a string type field, finds the records that have a number ending with the number that you enter. For example, searching a list of incidents by selecting Number and entering 4 shows records with numbers like INC0000004 and INC0000014.
- The Search option appears when you select for text. It returns records that contain the search term in any field.

**Note:** In UI11, the Go to and Search options are displayed side-by-side in the header.

1. Navigate to a list of records.
2. If necessary, personalize the list to add the desired fields.
3. Select a field or select for text.
4. Enter the search text. Use wildcards to further refine list searches.
5. Press Enter to execute the search.

In UI11, you can press Enter or click the search icon (🔍).

Search by one or more columns in a list

In addition to searching by field or for text, you can search by one or more individual columns in a UI16 or UI15 list.

Role required: none

---

**Figure 39: Column search**

This search supports queries that include AND, but does not support OR.

1. Click the search icon (🔍) to expand the column headers and add a search field to each column.
2. To search a single column, enter the search text in the desired column search field and press the Enter key.
   Use wildcards to further refine column searches. For example, use the * to define a contains search.
3. To search multiple columns, perform one of the following actions.
   - Enter the search text in each of the desired column search fields and press the Enter key.
• Search an individual column and then search additional columns based on the results of the first search.

The search returns records that match the search term.

To clear a column search, complete one of the following actions.
• Delete the text in the search field for the desired column and press the Enter key. This returns results for any remaining column search criteria.
• Delete the text in all the column search fields to return all records in the list.

Available list search wildcards

The platform supports several wildcard characters to expand and refine search results.

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*search-term</td>
<td>Search for values that contain search-term.</td>
</tr>
<tr>
<td>%search-term%</td>
<td></td>
</tr>
<tr>
<td>%search-term</td>
<td>Search for values that end with search-term.</td>
</tr>
<tr>
<td>search-term%</td>
<td>Search for values that start with search-term.</td>
</tr>
<tr>
<td>=search-term</td>
<td>Search for values that equal search-term.</td>
</tr>
<tr>
<td>!*searchterm</td>
<td>Search for values that do not contain search-term.</td>
</tr>
<tr>
<td>!=searchterm</td>
<td>Search for values that do not equal searchterm.</td>
</tr>
</tbody>
</table>

Searching without wildcard characters

If you enter text in the search box without using a wildcard, the search is performed for values greater than or equal to the value you enter. For text data-type-fields, this means that the search first sorts the records on the selected field, then finds the first record that starts with the text and all following records. For numeric data-type-fields, this means that the search finds all records where the number field ends with the entered number.

Note: The system treats some string fields that contain record numbers as numeric fields. Any field named number or u_number is treated as a numeric field.

Setting default search behavior

Administrators can add a property to perform a default contains search instead of a greater than search. To make this change, Add the property glide.ui.goto_use_contains and set the property Value to true.
Sort a list

You can quickly find information in a list by sorting the list. The method that the system uses to sort a list depends on the type of record in the sort column.

Role required: none

Consider the example record: Label=Requested Item; Value=sc_req_item. Field types other than choice list types are sorted based on the label. The example record label, Requested Item, would be sorted with records whose labels start with "r."

A choice list field lets the user select from a pre-defined set of choices. Choice list fields are sorted by the underlying dictionary entry value of the field, not by the label. The example record value, sc_req_item would be sorted with records whose values start with "s."

Sorting by value can be useful. For example, choices for the State of tasks are ordered New > Work in Progress > Closed. Based on the label, the sort would be Closed > New > Work in Progress.

The current sort order is indicated by an arrow next to the column name. A downward pointing arrow indicates the column is sorted in descending order. Only the primary sort order is indicated.

**Note:** The following sorting rules apply to lists:
- In a translated instance, only translated_text, translated_field, and translated_html type columns support sorting.
- The column label should be created in English first then translated accordingly.
- You cannot sort a list by an array-based field, such as a Glide list.

To sort a list, use one of the following methods:
- Click a column name to sort the list in ascending order. Click again to sort in reverse order.
- Right-click a column name and select Sort (a to z) or Sort (z to a) to sort in ascending or descending order, respectively.
- Specify a sort order with a filter. Filters provide for sorting by more than one column (for example, by Category and then Subcategory).

Grouped lists

Grouping aggregates a list by a field and displays the record count per group. Grouping can help you find data quickly by organizing and providing a summary of search or filter results.

For example, this picture shows active incidents grouped by caller.
Use a grouped list

You can quickly find information in a list by grouping the list. There are many ways you can use a list after you group the records in it.

Role required: none

1. To group items in a list, use one of the following methods.
   - On the title bar, click the context menu and select Group By. Select the field by which to group the list. To remove a grouping, select -- None --.
   - Right-click the column name and select Group By. To remove a grouping, select Ungroup.

2. Use any of the following actions to expand the groups.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To expand or collapse a group</td>
<td>Click the arrow (↑) next to the group name.</td>
</tr>
<tr>
<td>To expand or collapse all groups</td>
<td>Click the arrow at the top of the list.</td>
</tr>
<tr>
<td>To open the full list for a group</td>
<td>Click the group name.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td><strong>To see all records for a given group</strong></td>
<td>Open the full list.</td>
</tr>
</tbody>
</table>

3. To change the number of records per page, click the title menu and select **Show**.

The maximum number of records shown per group is the *number of records per page* in list view.

**Methods for list edits**

Users can edit data in lists using various methods.

**Quick edit functions**

To edit a record in a list using quick edit functions, right-click a field and select the appropriate function.

- **Assign to me**: For records that use assignments, places the logged-in user's name into the **Assigned to** field.
- **Approve**: For records that use approvals, changes the record's approval state to **Approved**.
- **Reject**: For records that use approvals, changes the record's approval state to **Rejected**.
- **Assign tag**: For records you want to track based on a user-defined label, lets you select an existing tag or add a tag.

**List editor**

The list editor lets you edit field values in a list without opening a form. Administrators can configure the list editor. By default, list editing is disabled for some tables. Fields of *certain types* cannot be edited from lists.

**Multiple record edits**

You can edit more than one record at the same time using the list editor or an editing form.

**Use the list editor**

The list editor allows users to edit field values directly from a list without navigating to a form.

Role required: none

The list editor lets you edit field values in a list without opening a form. Administrators can configure the list editor. By default, list editing is disabled for some tables. Fields of *certain types* cannot be edited from lists.

Before the list editor opens, access rights to edit the field are verified. A loading indicator appears if this process takes longer than expected. If the field has a dependency relationship (for example, **Category** and **Subcategory**), then a composite editor opens to allow editing of all dependent fields. You must have rights to edit all dependent fields to use the list editor.

1. Double-click in an empty area of the field. The appropriate editor for the field type opens.
2. Enter the appropriate values and click Save (✓), or click Cancel (✗) to retain the original value.

3. To use keyboard navigation, press the Tab key until the first field in the list is selected, and then select the field to edit in one of the following ways.
   • Move right: Tab or the Right Arrow key.
   • Move left: Shift + Tab or the Left Arrow key.
   • Move down: the Down Arrow key.
   • Move up: the Up Arrow key.

   To select multiple fields in the same column, hold Shift and press the Down Arrow or the Up Arrow key.

4. Press the Enter key to open the list editor.

5. Enter a new value. To add another line in a multi-line text field, press Shift + Enter.

6. Save or cancel your changes in one of the following ways:
   • Press the Enter key. The new value is saved and the field below the edited field becomes selected.
   • Press the Tab key. If the list is configured to save immediately, the new value is saved. If the list is configured to save data by rows, an indicator appears beside the value and the list editor opens for the next field. The row is saved only when you navigate away from the row or click the check mark icon beside the row.
   • Press Ctrl + Enter keys. If the list is configured to save immediately, the new value is saved. If the list is configured to save data by rows, an indicator appears beside the value and the current field remains selected.
   • Press the Esc key. The list editor closes without saving changes and the field remains selected.

---

**Note:** Certain browsers use different key combinations to edit certain field values. For example, to edit a choice list using Chrome, press the Spacebar.

---

*Create a new record*

Users can create new records in list view. Administrators can configure the list editor to enable this feature for lists.
Roler required: none

1. Navigate to the empty row at the bottom of the list.
2. Open the list editor for a field in the row and enter a value.
3. Save or cancel your changes.

Figure 42: Insert a new row (record)

Configure the list editor
You can personalize how the list editor behaves.

Role required: none
1. Open the list.
2. Click the personalize list icon (⚙️) in the upper left corner.
   • To allow the list editor to open for the list, select the Enable list edit check box. Clear the check box to prevent the list editor from opening for the list.
   • To open the list editor with a double-click, select the Double click to edit check box. Clear the check box to open the list editor using a single click.
3. Click OK.

Edit multiple records in a list using the list editor
You can edit multiple records at the same time using the list editor. If you want to update a single field on multiple records to have the same value, the list editor is the quickest method.

Role required: none
1. Select the records you want to edit by performing one of the following actions.
   • To select multiple consecutive fields, hold Shift and drag in the desired fields, or select a cell and then press Shift + Up Arrow or Shift + Down Arrow.
   • To select multiple non-consecutive fields, press Shift and click in one of the desired fields, then hold Shift + Ctrl (Shift + Command on Mac), and click in the desired fields.
2. Open the list editor by double-clicking (or clicking, depending on setup) in an empty area of the field.
   The number of selected rows that you are editing is indicated. If any rows cannot be edited due to security constraints, that is indicated.
3. Enter the appropriate values and click Save.

**Edit multiple records in a list using an editing form**

You can edit multiple records at the same time using an editing form. If you want to edit multiple fields or fields that do not appear in the list view, use an editing form.

Role required: list_updater or admin

1. Select the records you want to edit by performing one of the following actions.
   - Select the check boxes in the record rows.
   - Select a field in the record rows.
   - Filter the list to show only the rows you want to edit.

2. Right-click the column header and select Update Selected or, if you filtered the data, select Update All.
   - An editing form opens.

3. Enter appropriate values in any of the fields and click Update to save your changes in all selected records.

**Personal lists**

Personal lists modify a specific list view according to individual preferences.

Personal list customizations do not affect what other users see in their lists. Administrators can manage the personal lists function.

Users can create personal lists and define the following layout options.

- Column selection and order
- Row spacing
• Text wrapping
• List editing
• List highlighting

Figure 44: UI16/UI15 personalize list
Personalize a list

You can personalize a specific list view based on individual preferences.

1. Open the list.
2. Click the personalize list icon (⚙️) in the upper left corner.
3. Use the slushbucket to select the columns and the desired order.

Figure 45: UI11 personalize list
The first non-reference field automatically links to the form view of the record. For this reason, consider using the record number as the first column in your personal list layout.

4. Select display options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display long text on more than one line</td>
<td>Select the <strong>Wrap column text</strong> check box. Clear the check box to display text on one line.</td>
</tr>
<tr>
<td>Condense the vertical space between rows</td>
<td>Select the <strong>Compact rows</strong> check box. Clear the check box to use standard row spacing.</td>
</tr>
<tr>
<td>Highlight list rows as the cursor passes over them</td>
<td>Select the <strong>Active row highlighting</strong> check box. Clear the check box to restore the static, alternate row highlighting.</td>
</tr>
<tr>
<td>Use updated field status indicators available in UI15 and UI16</td>
<td>Select the <strong>Modern cell coloring</strong> check box. Clear the check box to use field status indicators available in UI11.</td>
</tr>
</tbody>
</table>

5. Select list editing options (requires you to configure the list editor).

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow the list editor to open for the list</td>
<td>Select the <strong>Enable list edit</strong> check box. Clear the check box to prevent the list editor from opening for the list.</td>
</tr>
<tr>
<td>Open the list editor with a double-click</td>
<td>Select the <strong>Double click to edit</strong> check box. Clear the check box to open the list editor using a single click.</td>
</tr>
</tbody>
</table>

6. Click OK.

If a list is personalized, an indicator (☁️) appears in the upper left corner.

To reset a list to the default layout, click the personalize list icon and select the **Reset to column defaults** check box.

**Forms**

A form displays information from one record in a data table.

The specific information depends on the type of record displayed. Users can view and edit records in forms. Administrators can configure what appears on forms.
Figure 46: UI16/UI15 form elements

Form header

Embedded list

Fields

Sections

Related links

Related lists
### Table 27: Form elements

<table>
<thead>
<tr>
<th>Form element</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form header</td>
<td>Provides navigation tools and actions related to the record.</td>
</tr>
<tr>
<td>Fields</td>
<td>Store specific data about the record.</td>
</tr>
<tr>
<td>Sections</td>
<td>Group related information on the form. Users can collapse or display sections using tabs.</td>
</tr>
<tr>
<td>Related links</td>
<td>Provide access to additional functions based on record type and system setup. Administrators can add related links to forms using UI actions.</td>
</tr>
<tr>
<td>Related lists</td>
<td>Display records in other tables that have relationships to the current record.</td>
</tr>
<tr>
<td>Embedded lists</td>
<td>Allow for editing related lists without having to navigate away from the form. Changes are saved when the form is saved.</td>
</tr>
<tr>
<td>Response time indicator</td>
<td>Appears at the bottom of some forms to indicate the processing time required to display the form.</td>
</tr>
</tbody>
</table>

### Form headers for UI versions

Each UI version, UI16, UI15, and UI11, has a different form header that offers different navigation icons.

To learn about form header navigation, select the form header topic that corresponds to the UI version your organization uses.

**UI16 form header** on page 98

![UI16 form header](image)

**UI15 form header** on page 100

![UI15 form header](image)

**UI11 form header** on page 101

The UI11 form header can have a blue background, but the icons are as pictured in this image.

![UI11 form header](image)

**UI16 form header**

The form header displays the title of the form (table) and provides a number of controls.
The UI16 form header includes the following controls.

**Figure 47: Form header UI16**

<table>
<thead>
<tr>
<th>Control</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back</td>
<td>🔄</td>
<td>Navigates to the previously viewed page without saving changes.</td>
</tr>
<tr>
<td>Form context menu</td>
<td>📚</td>
<td>Appears when a user clicks the menu icon beside the form title or right-clicks the form header. This is also called a right-click menu.</td>
</tr>
<tr>
<td>Attachments</td>
<td>📂</td>
<td>Allows users to view and add attachments to the record. For more information, see <a href="#">Add attachments</a> on page 127.</td>
</tr>
<tr>
<td>Show activity stream</td>
<td>📰</td>
<td>Moves focus to the journal entry section of the form.</td>
</tr>
<tr>
<td>Personalize form</td>
<td>🛠️</td>
<td>Opens the form personalization menu. For more information, see <a href="#">Personalize a form</a> on page 126.</td>
</tr>
<tr>
<td>More options</td>
<td>📟</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>⇐</td>
<td>Opens an email window.</td>
</tr>
<tr>
<td>Control</td>
<td>Icon</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Template bar</td>
<td><img src="image" alt="Icon" /></td>
<td>Opens a bar at the bottom of the form and lists available templates.</td>
</tr>
<tr>
<td>Annotations</td>
<td><img src="image" alt="Icon" /></td>
<td>Displays on-form annotations. For more information, see Form annotation on page 716.</td>
</tr>
<tr>
<td>Tags</td>
<td><img src="image" alt="Icon" /></td>
<td>Displays the option to create custom tags and categorize documents.</td>
</tr>
<tr>
<td>Remaining controls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submit or Update</td>
<td></td>
<td>Saves changes and returns to the previously viewed page.</td>
</tr>
<tr>
<td>Related actions</td>
<td></td>
<td>Provides standard actions in the form header for some tables, such as Close Incident or Resolve Incident in incident.</td>
</tr>
<tr>
<td>Delete</td>
<td></td>
<td>Deletes the record from the list.</td>
</tr>
<tr>
<td>Previous and Next</td>
<td><img src="image" alt="Icon" /></td>
<td>Opens the previous and next record on the list from which the record was accessed.</td>
</tr>
</tbody>
</table>

UI15 form header

The form header displays the title of the form (table) and provides a number of controls. The UI15 form header includes the following controls.

![Image of UI15 form header]

Figure 48: Form header UI15
Table 29: UI15 form header controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back</td>
<td><img src="image" alt="Back Icon" /></td>
<td>Navigates to the previously viewed page without saving changes.</td>
</tr>
<tr>
<td>Form context menu</td>
<td><img src="image" alt="Menu Icon" /></td>
<td>Appears when a user clicks the menu icon beside the form title or right-clicks the form header. This is also called a right-click menu.</td>
</tr>
<tr>
<td>Edit tags</td>
<td><img src="image" alt="Edit Tag Icon" /></td>
<td>Displays the <strong>Add tag</strong> field below the form header, which allows users to create custom tags and categorize documents. For more information, see <a href="#">Tags</a> on page 166.</td>
</tr>
<tr>
<td>Attachments</td>
<td><img src="image" alt="Attachment Icon" /></td>
<td>Allows users to view and add attachments to the record. For more information, see <a href="#">Add attachments</a> on page 127.</td>
</tr>
<tr>
<td>Annotations</td>
<td><img src="image" alt="Annotation Icon" /></td>
<td>Displays on-form annotations. For more information, see <a href="#">Form annotation</a> on page 716.</td>
</tr>
<tr>
<td>Email</td>
<td><img src="image" alt="Email Icon" /></td>
<td>Opens an email window.</td>
</tr>
<tr>
<td>Show activity stream</td>
<td><img src="image" alt="Activity Stream Icon" /></td>
<td>Opens the document feed for this record in a flyout window.</td>
</tr>
<tr>
<td>Personalize form</td>
<td><img src="image" alt="Personalization Icon" /></td>
<td>Opens the form personalization menu. For more information, see <a href="#">Personalize a form</a> on page 126.</td>
</tr>
<tr>
<td>Submit or Update</td>
<td><img src="image" alt="Submit or Update Icon" /></td>
<td>Saves changes and returns to the previously viewed page.</td>
</tr>
<tr>
<td>Related actions</td>
<td><img src="image" alt="Related Actions Icon" /></td>
<td>Provides standard actions in the form header for some tables, such as Close Incident or Resolve Incident in incident.</td>
</tr>
<tr>
<td>Delete</td>
<td><img src="image" alt="Delete Icon" /></td>
<td>Deletes the record from the list.</td>
</tr>
<tr>
<td>Previous and Next</td>
<td><img src="image" alt="Previous and Next Icon" /></td>
<td>Opens the previous and next record on the list from which the record was accessed. These controls save changes to the record.</td>
</tr>
</tbody>
</table>

**UI11 form header**

The form header displays the title of the form (table) and provides a number of controls.
The form header for UI11 includes the following controls.

![Image of UI11 form header]

**Figure 49: UI11 form header**

<table>
<thead>
<tr>
<th>Control</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back</td>
<td><img src="arrow_icon.png" alt="Arrow Icon" /></td>
<td>Navigates to the previously viewed page without saving changes.</td>
</tr>
<tr>
<td>Form context menu</td>
<td><img src="down_arrow_icon.png" alt="Down Arrow" /></td>
<td>Appears when a user points to the arrow beside the form title or right-clicks the form header.</td>
</tr>
<tr>
<td>Submit or Update</td>
<td><img src="up_arrow_icon.png" alt="Up Arrow" /></td>
<td>Saves changes and returns to the previously viewed page.</td>
</tr>
<tr>
<td>Related actions</td>
<td></td>
<td>Provides standard actions in the form header for some tables, such as <strong>Close Incident</strong> or <strong>Resolve Incident</strong> in incident.</td>
</tr>
<tr>
<td>Delete</td>
<td><img src="trash_can_icon.png" alt="Trash Can" /></td>
<td>Deletes the record from the list.</td>
</tr>
<tr>
<td>Attachments</td>
<td><img src="attachment_icon.png" alt="Attachment Icon" /></td>
<td>Allows users to view and add attachments to the record.</td>
</tr>
<tr>
<td>Email</td>
<td><img src="email_icon.png" alt="Email Icon" /></td>
<td>Opens an email window.</td>
</tr>
<tr>
<td>Show live feed</td>
<td><img src="live_feed_icon.png" alt="Live Feed Icon" /></td>
<td>Opens the document feed for this record in a flyout window.</td>
</tr>
<tr>
<td>Previous</td>
<td><img src="up_arrow_icon.png" alt="Up Arrow" /></td>
<td>Opens the previous record on the list from which the record was accessed. This control saves changes to the record.</td>
</tr>
<tr>
<td>Next</td>
<td><img src="down_arrow_icon.png" alt="Down Arrow" /></td>
<td>Opens the next record on the list from which the record was accessed. This control saves changes to the record.</td>
</tr>
<tr>
<td>Toggle tabs</td>
<td><img src="tabs_icon.png" alt="Tabs Icon" /></td>
<td>Toggles between tabbed and sequential arrangements of form sections and related lists.</td>
</tr>
<tr>
<td>Control</td>
<td>Icon</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Collapse all sections</td>
<td></td>
<td>Collapse all form sections.</td>
</tr>
<tr>
<td>Expand all sections</td>
<td></td>
<td>Expand all form sections.</td>
</tr>
<tr>
<td>Collapse</td>
<td></td>
<td>Collapse the main form section.</td>
</tr>
<tr>
<td>Expand</td>
<td></td>
<td>Expand the main form section.</td>
</tr>
</tbody>
</table>

**Form context menu**

The form context menu provides controls based on the table and user access rights. Administrators can customize some of the options available on a context menu using UI actions.

Access the form context menu by clicking an icon beside the form title or by right-clicking the form header.

- UI16 and UI15 menu icon (_gateway_global_icon_0009)
- UI11 arrow icon
Figure 50: UI16 form context menu

The form context menu includes the following options.

**Note:** Some of the options displayed on the form context menu depend on the user role, installed applications, and version of the UI.

### Table 31: Form context menu options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>Saves changes without leaving the form view.</td>
</tr>
<tr>
<td>Related actions</td>
<td>Provides standard actions in the form context menu for some tables, such as Add to Visual Task Board or Create Change in incident.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Insert</td>
<td>Saves the data as a new record and returns to the previously viewed page. For more information, see <em>Insert a record</em> on page 123.</td>
</tr>
<tr>
<td>Insert and Stay</td>
<td>Saves and displays the new record. For more information, see <em>Insert a record</em> on page 123.</td>
</tr>
<tr>
<td>Configure</td>
<td>Provides administrative functions, such as configuring forms.</td>
</tr>
<tr>
<td>Templates</td>
<td>(UI11 only) Provides administrative functions, such as creating templates. For more information, see <em>Save a form as a template in UI11</em> on page 730.</td>
</tr>
<tr>
<td>Toggle Template Bar</td>
<td>(UI15 only) Shows or hides the template bar. For more information, see <em>Toggle the template bar</em> on page 726.</td>
</tr>
<tr>
<td>Export</td>
<td>Exports data to PDF. Administrators can also export to XML. For more information, see <em>Export data</em> on page 1494.</td>
</tr>
<tr>
<td>View</td>
<td>Changes fields to a predefined layout. Note that switching views submits the form, which saves all changes. Administrators can customize views.</td>
</tr>
<tr>
<td>Assign Tag</td>
<td>Allows the user to assign a new or existing tag to a record, which provides quick access to frequently referenced or urgent information. When tag is assigned to a record, the record is displayed on the Tagged Documents page. For more information, see <em>Tags</em> on page 166.</td>
</tr>
<tr>
<td>Copy URL</td>
<td>Copies to the clipboard the URL for the form view of the record. Follow browser instructions if browser security measures restrict this function.</td>
</tr>
<tr>
<td>Copy sys_id</td>
<td>Administrators only. Copies to the clipboard the sys_id of the record. Follow browser instructions if browser security measures restrict this function.</td>
</tr>
<tr>
<td>Show XML</td>
<td>Administrators only. Displays record data in XML format.</td>
</tr>
<tr>
<td>History</td>
<td>Administrators only. Displays audit history for the record, which must be enabled for the table. For more information, see <em>Enable auditing for a table</em> on page 2563.</td>
</tr>
<tr>
<td>Reload Form</td>
<td>Reloads information from the database to refresh the form view.</td>
</tr>
</tbody>
</table>

Form fields

A field represents an individual item of data on a record.
Users can view and modify field data on a form.

**UI16 and UI15 field status indicators**

Indicators are used on some fields to denote a special field type.

A field status indicator is a colored asterisk that may appear to the left of mandatory fields. Field status indicators change colors to represent different states of mandatory fields.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unpopulated</td>
<td>Required field is empty. The user must enter a value to save the form. Default color is red.</td>
</tr>
<tr>
<td>Populated - saved</td>
<td>Required field contains a value that was saved or needs to be saved. Default color is grey.</td>
</tr>
<tr>
<td>Populated - unsaved</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 51: UI16/UI15 field status indicators**

**Note:** Tabs containing required fields have black asterisks.

**UI11 field status indicators**

A field status indicator is a colored bar that may appear to the left of form elements.

These indicators convey the following statuses.

<table>
<thead>
<tr>
<th>Field status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Required field that is empty. The user must enter a value to save the form. Default color is red.</td>
</tr>
<tr>
<td>Populated mandatory</td>
<td>Required field for which a value has already been saved. Default color is light red. If the user enters a new value, the field status indicator changes to modified (default green).</td>
</tr>
</tbody>
</table>
Field status | Description
-------|-------
Modified | Contains data that has not been saved. Default color is green.
Read-only | User cannot edit on the form. Default color is orange.

Figure 52: UI11 field status indicators

An administrator can customize the color of these field status indicators by navigating to System Properties > CSS:

![System Properties](image)

Figure 53: Customize UI11 field status indicator colors

Configure the activity filter

The activity formatter header contains a filter that lets users select which of the available fields to show in the activity list.

Role required: none

All fields on the activity list display in alphabetical order.

Configure the activity filter in UI16 and UI15

1. Open the filter by clicking the activity filter icon (🔍) in the Activity header.
2. Select the fields for which you want to display activity.
Fields are added or removed dynamically as the selection is made. In UI16 the All option does not display.

3. Click the activity filter icon again to close the filter.

Configure the activity filter in UI11

1. Open the filter by clicking the show/hide filter icon (AsStream) in the Activity header.
2. Select the fields for which you want to display activity. Fields are added or removed dynamically as the selection is made.
3. Click the show/hide filter icon again to close the filter.
   A filter icon appears on the Activity header to indicate that a filter is in effect.
4. Click the filter icon to toggle the filter on (display only the selected fields) or off (display all fields).
   When the activity filter is on, the filter icon has a blue border. The activities list shows the history of only those fields selected in the filter and the Activity header indicates how many of the available fields are selected, such as (4 of 5).
   When the activity filter is off, the blue highlight is removed from the filter icon and the form shows all available fields. If all fields are checked, the filter icon is not displayed.

Form sections

Form sections provide headers for a group of related fields.

Use these icons to work with form sections.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>(UI15) (previous Ul)</td>
<td>Expands a form section.</td>
</tr>
<tr>
<td>(UI15) (previous Ul)</td>
<td>Collapses a form section.</td>
</tr>
<tr>
<td>(UI11)</td>
<td>Toggles between a tabbed and sequential arrangement of sections.</td>
</tr>
</tbody>
</table>

**Note:** This icon is removed from the form header in UI15 and UI16. To enable or disable form tabs, click the gear icon in the form header and toggle the Tabbed forms option.

Embedded lists

Some forms may show related lists as embedded. Changes to embedded lists are saved when the form is saved.

Use these controls to work with an embedded list. For more information, see Edit a form on page 121.
### Table 34: Working with embedded lists

<table>
<thead>
<tr>
<th>Task</th>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expand an embedded list</td>
<td>![+]</td>
<td>Click the expand icon in the list header.</td>
</tr>
<tr>
<td>Collapse an embedded list</td>
<td>![−]</td>
<td>Click the collapse icon in the list header.</td>
</tr>
<tr>
<td>Insert a new row</td>
<td>![ ]</td>
<td>Double-click <strong>Insert a new row...</strong></td>
</tr>
<tr>
<td>Edit a row</td>
<td>![ ]</td>
<td>Double-click in an empty area of that field. See <em>Use the list editor</em> on page 90.</td>
</tr>
<tr>
<td>Delete a row</td>
<td>![X]</td>
<td>Click the delete icon beside the row. New rows are removed immediately. Existing rows are designated for deletion when the record is saved. To clear this designation, click the delete icon again.</td>
</tr>
</tbody>
</table>
Related lists

Related lists appear on forms and show records in tables that have a relationship to the current record. Users can view and modify information in related lists like any other list. Administrators can configure related lists to appear on forms and in hierarchical lists. Related lists do not have a size limit.

Select related records

You can select records for a related list.

Role required: none

1. Open the record for which you are selecting related records, such as a problem record.
2. Locate the related list, for example, Incidents.
3. Click Edit.
4. Using the slushbucket, select the records to associate with the current record. When you select a record, information about it appears below the Collection list to help you identify the record to select.
5. Click **Save**.

### Create a new related record

You can add a new record to the database from a related list.

**Role required:** You must have at least one role that lets you create a record in the related table.

1. Open the record to which you are adding a related record, for example, a change request.
2. Locate the related list, for example, **Change Tasks**.
3. Click **New**. A new form for the related table opens, for example, the Change Task form.
4. Complete the form and click **Submit**. The new record is added to the related table and to the related list of the record.

### Configure when a related list loads

If there are many related lists on a form or many records in the related lists, the form may load slowly. In UI16 and UI15, you can improve form response times by configuring related lists to load manually on demand or automatically after the rest of the form loads.

**Role required:** none

1. Click the settings icon (⚙️) in the banner frame to open the system settings menu.
2. Select one of the following options under **Related List Loading**.
   - **With the Form**: Related lists load as soon as you open the form. This is the default setting.
   - **After Form Loads**: Related lists load after the rest of the form loads.
   - **On-demand**: Related lists load on demand. When this option is selected, a **Load Related Lists** button appears at the bottom of each form that contains related lists. You must click the button to load related lists.
3. Close the system settings menu.

### Create a default filter

You can create a default filter that is applied to a related list when a form loads.

**Role required:** admin

1. Navigate to the related list (example, **Incidents** on a problem record).
2. Create the desired filter using the condition builder (example, `[Active] [is] [true]`).
3. Click **Set as Default Filter**.

Configure the edit option

You can configure the edit option that allows users to add records to related lists in forms.

Role required: admin

The **Edit** button is available for related lists that represent many-to-many and one-to-many relationships. Defined related lists may not have an **Edit** button as related records are added to these lists automatically based on the relationship between the records.

1. Navigate to the related list, such as the **Incidents** related list on a problem record.
2. Right-click any column heading and select **Configure > List Control**.
3. Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show the Edit option</td>
<td>Clear the <strong>Omit edit button</strong> check box.</td>
</tr>
<tr>
<td>Hide the Edit option</td>
<td>Select the <strong>Omit edit button</strong> check box.</td>
</tr>
<tr>
<td>Show the Edit option if the check box is cleared and</td>
<td>Click <strong>Enable Edit</strong>.</td>
</tr>
<tr>
<td>the button does not appear</td>
<td></td>
</tr>
</tbody>
</table>

4. If the list represents a one-to-many relationship (example, incidents can be associated with only one problem, but problems can be associated with many incidents), complete the following steps.
   a) Open a record in the target table (example, Incident).
   b) Right-click the header and select **Configure > Dictionary**.
   c) Open the dictionary entry for the reference field (example, **Problem ID** on **Incident**).
   d) In the Reference Specification section, select the **Reference floats** check box.
5. Click **Update**.

Create defined related lists

You can add default related lists to the form for all users to see when viewing records.

Role required: admin

For example, you may include a list of related incidents at the bottom of a problem record, or a list of members at the bottom of a group record. This functionality is dependent on reference fields or many-to-many table relationships. If two tables are related via the system dictionary, one can appear as a related list on the other.

Defined related lists allow relationships between arbitrary tables to be expressed as a related list. Any two tables that can have a logical relationship can appear as a parent/child pair via a related list. The following are some examples.

- On an incident record, show all incidents opened by the same caller.
- On a user record, show the last 20 transactions that user has made.
- On an incident record, show all problems opened on the reported CI.
These relationships are beyond the relationships normally defined in the system dictionary through reference fields and many-to-many relationships.

Every related list requires a relationship record. Before creating a new relationship, verify that there is not an existing relationship record that already provides the needed information. Use the following steps to create a new relationship record.

1. Click System Definition > Relationships.
2. Click New.
3. Specify the relationship record fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type the name of the related list. The form configuration page displays this name in the list of available lists.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use JavaScript in place of the Applies to table and Queries from table fields.</td>
</tr>
<tr>
<td>Applies to table</td>
<td>Select the table on which the related list appears. This field is hidden in Advanced relationship records.</td>
</tr>
<tr>
<td>Queries from table</td>
<td>Select the table from which this related list retrieves data. This field is hidden in Advanced relationship records.</td>
</tr>
<tr>
<td>Apply to</td>
<td>Type a script to specify the table on which the related list appears. This field is only visible with the Advanced check box selected.</td>
</tr>
<tr>
<td>Query from</td>
<td>Type a script to specify the table from which this related list retrieves data. This field is only visible with the Advanced check box selected.</td>
</tr>
<tr>
<td>Query with</td>
<td>Type a script to specify the records to include from the table the relationship queries.</td>
</tr>
<tr>
<td>Insert Callback</td>
<td>Type a script to run after a successful insert action. This field is only visible with the Advanced check box selected.</td>
</tr>
</tbody>
</table>

4. Click Submit.

Add incidents by same caller related list
This example adds the existing relationship, Incidents by Same Caller, to incident forms.

- **Applies to table**: This list appears as an available related list when viewing an incident record.
- **Queries from table**: This list displays a list of incidents.
- **Query with**: This script selects records where the caller_id matches the caller_id of the parent record (the incident you are viewing).  

© 2017 ServiceNow. All rights reserved. 114
1. Open an incident.
2. Right-click the header and select **Configure > Related Lists**.
   The name of the existing relationships appear in your list of available lists.
3. Move the **Incidents by Same Caller** list to the **Selected** list.
4. Click **Save**.
| Number  | INC0000052          | Caller       | Bud Richman | Category | Software | Subcategory | -- None -- | Affected CI | SAP Financial Accounting | Impact   | 1 - High                  | Urgency   | 1 - High                  | Priority   | 1 - Critical               | Assigned to | Fred Lundy                  | Assigned group | Software |
|---------|---------------------|--------------|-------------|----------|----------|-------------|------------|-------------|-------------------------|----------|--------------------------|-----------|--------------------------|------------|---------------------------|-------------|---------------------------|
| Short description | SAP Financial Accounting application appears to be down | **Incidents by Same Caller** (4) | **New** | Go to | Number | Search | < | < | < | < | < | < | < | < | < | < | < | < | < | < |
| INC0000052 | Bud Richman | SAP Financial Accounting application appears to be down | Software | 1 - Critical | Active | Software |
| INC0000046 | Bud Richman | Can't access SFA software | Software | 3 - Moderate | New | Software |
| INC0000049 | Bud Richman | JavaScript error on hiring page of corporate website | Inquiry / Help | 3 - Moderate | Awaiting | Problem |
Notes and limitations:

• If you click the **New** button on one of these new relationships, the system attempts to ensure that the new record matches the list conditions. For example, clicking **New** on our example list results in an incident where the **caller_id** is pre-populated with **Bud Richman**.

• When scripting your condition, **current** is the record to which you want to add queries while **parent** is the main record being displayed.

• These relationships do not refresh until you update a form. In the example, if you changed the caller from **Bud Richman** to **Fred Luddy**, the list at the bottom of the screen still displays Bud Richman's incidents until you save the incident.

• You are not limited to a single query condition. It is, for example, possible to have a related list of all incidents opened by the same caller in the last week, or all open incidents opened by the same caller.

• The **current** and **parent** objects cannot be used with the **Queries from** field. Instead, the **gs** object is available for GlideSystem calls.

• The **Edit** button is not available on defined relationships, as the relationship is completely scripted.

---

**Add transactions by user related list**

This example adds a new relationship, Last 20 transactions, to the user form.

Role required: admin

The new relationship creates a list of task records opened by the current user.

1. Navigate to **System Definition > Relationships**.
2. Click **New**.
3. Create a new relationship with the following values.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Last 20 transactions</td>
</tr>
<tr>
<td>Applies to table</td>
<td>User [sys_user]</td>
</tr>
<tr>
<td>Queries from table</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Queries with</td>
<td><code>current.addQuery('opened_by', parent.sys_id);</code></td>
</tr>
</tbody>
</table>

4. Click **Submit**.
5. Navigate to **User Administration > Users**.
6. Select any user.
   For example, select David Loo.
7. Configure the related lists for the user form and add the related list **Last 20 transactions**.
8. Navigate to **User Administration > Users**.
9. Select any user.
   For example, select David Loo.
10. The **Last 20 transactions** related list displays all task records opened by the current user.

Notes and limitations:

• This query returns all task records opened by the user. The related list defaults to displaying 20 task records per page.

• The parent variable refers to the **Applies to table**. In this case, it applies to the User [sys_user] table.

• Querying against a record sys_id is a common query method for relationships.
Add fields to selections in a related list

When you click Edit in a related list and select an item, information about the item appears below the list. You can expand the fields that appear for the item to provide more information.

Role required: admin
Figure 55: Fields listed for selected record
To add fields to selections in a related list, record the exact field name, or element, not the label name. This example demonstrates adding fields to the Groups related list [sys_user_group] as illustrated.

1. Navigate to System UI > Views.
2. Select sys_ref_list.
3. To add fields from the Group table, in the Lists related list, select sys_user_group.
4. In the List Elements related list, click New.
5. Enter the field name and its relative position when it appears in the description.
   Fields appear from top to bottom, lowest to highest number.
6. Click Submit.

Create default filter for a related list

You can create a default filter or reference qualifier to filter the available options for related lists.

Role required: personalize_list

A filter is used to restrict the records that are listed in the slushbucket that displays when a user clicks Edit in a related list.
Figure 56: Unfiltered list of records

1. Right-click a related list column header and click Configure > List Control
2. Configure the form to add the Edit default filter field, if necessary.
3. Using the condition builder, create a default filter.
4. Click Update.
   When users click the Edit button in the related list, the list of records they can select is filtered according to the default filter.

Edit a form

You can edit a record in the form view.
Role required: none

1. Navigate to the form.

   If another user is looking at the same record, you see their avatar with a green dot in the form header. If more than one user is viewing the record, the avatar is a number, and you can point your cursor to it to see the names.

2. Enter appropriate data.

   **Note:** Depending on system setup, the data entered in a form can affect other options on the form or change its appearance. For example, if you select a status of **Closed** for an incident, a required **Close Notes** field may appear. Administrators can create custom form effects in various ways, including UI policies and client scripts.

3. Save the record using one of these methods:
   - Click **Submit** or **Update** to save changes and return to the previously viewed page.
   - Right-click the form header and select **Save** to save changes without leaving form view.
   - Right-click the form header and select **Insert** or **Insert and Stay** to save the data to a new record instead of updating the current record. For more information, see **Insert a record** on page 123.

On UI16 forms, fields update dynamically and display a field update icon (短板) to indicate changes made by other users. You can point to the field update icon to see which user made the update. This dynamic indication helps to prevent users from accidentally overwriting each other’s work.

**Note:** If you entered a different value from the value saved by the other user, the system does not update the field automatically. Change your value manually to match value entered by the other user. Otherwise, you overwrite the change when you save the record.

---

© 2017 ServiceNow. All rights reserved.
Insert a record

Insert provides a method for creating multiple similar items, such as email notifications, users, groups, or business rules.

Role required: none

Administrators can enable the Insert function for task records. For more information, see Configure task record insert options on page 705.

1. Open a similar item.
2. Edit the item and give it a new Name.
3. Right-click the form header and select one of the following options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insert</strong></td>
<td>Saves the item as a new record and returns to the previously viewed page.</td>
</tr>
<tr>
<td><strong>Insert and Stay</strong></td>
<td>Saves and views the new record.</td>
</tr>
</tbody>
</table>

Apply a template in a form

Your organization may use templates to help with frequently requested tasks. A template contains prepopulated fields that default when the template is selected.

Role required: none

This option is not available in UI11.

1. Navigate to the form.
2. Do the appropriate action for your version of the UI.

<table>
<thead>
<tr>
<th>UI16</th>
<th>1. Click the more options icon (○○○) in the form header.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. Select <strong>Toggle Template Bar</strong>.</td>
</tr>
<tr>
<td>UI15</td>
<td>1. Right-click the form header.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
3. Select the template to apply.
   If there is a conflict between the selected template and the form, a conflict message appears with a link to view the details of the conflict. For example, if a field that does not appear on the form is populated in the template, the conflict message details identify the missing field and the value is not set. If you have a conflict message, review the details and notify your administrator if necessary.

Specific fields are populated with values from the template.

4. Complete the form.
   You can change any value that was populated from the template.

5. Click **Submit** to create the record.

### Cancel changes to a form

Cancel changes to a form by navigating away from the form without saving.

**Role required:** none

1. Navigate away from the form view without saving the record.

2. If a message appears, click **Leave this Page** to confirm cancellation.

   **Note:** Administrators can disable the confirmation message by setting the glide.ui.dirty_form_support property to false.

### Use form pane tabs

When using the split screen feature, you can open multiple forms in the form pane. Form pane tabs are available in UI15 only.

Form pane tabs appear above the form header. The current tab is highlighted.
1. View a list in the content frame, for example, the Incident list.
2. Click the List and Form View button on the Edge to split the screen. The list is displayed in the list pane on the left and an empty form pane with a new highlighted tab is displayed on the right.
3. Select an incident record in the list pane to display the incident in the highlighted tab. The incident number appears in the tab at the top of the form pane.
4. Click the + tab to add a tab to the form pane. The new tab is added to the right of the current tabs and has the generic name New Tab until you select another incident from the list pane.
5. To remove a tab, select the tab to be deleted and click the x.

Form personalization

When the form personalization feature is activated, users can personalize fields to appear on a specific form view according to individual preferences. Form personalization is available in UI16 and UI15.

Personal form customizations do not affect what other users see on their forms.

In contrast to configuring a form, personalizing a form does not enable users to perform the following actions:

- Change the order of fields on the form.
- Add fields that are not configured to appear on the form.
- Hide mandatory fields.

**Note:** Personalizing a form in this way modifies the form for you only. To make changes to a form that are visible to all users, you must configure the form.
Personalize a form

You can personalize fields to appear on a specific form view.

Role required: itil, personalize_form, or admin

1. Open any record.
2. In the form header, click the gear icon (⚙).
   The Personalize Form menu appears and hide field icons appear on the form.

   Fields you previously hid are indicated by a cleared gray check box.

   **Note:** Only the fields that the form is configured to display are available in the Personalize Form menu. To add fields to the form, you must configure the form layout.

3. Clear a check box or click the hide field icon (숨기기) by a field.
   Some check boxes are gray and selected. These check boxes indicate fields that cannot be hidden because they are required or are already hidden by UI policies or client scripts.
4. Click outside the Personalize Form menu to save your changes.
   If you navigate away from the form, the next time you personalize the form you must reset your customizations to show fields that you have hidden.
5. To restore the default form view, personalize the form and click **Reset**.
Add attachments

You can upload a file as an attachment to an incident, a knowledge article, a change request, or to another type of record.

Attachment file size is not limited by default. However, uploading large attachments can cause issues with the user's active session on the instance. Empty attachments (file size of 0 kb) are not supported.

Note: Do not use the Upload File module in the System Definition application. It is not compatible with multi-node instances.

1. Navigate to the record (example, an incident record).
2. Click the attachments icon ( ).
3. Click Choose Files or Browse, depending on your browser, and navigate to a file.
4. Optional: Upload multiple files in one of the following ways.
   - Select multiple files at the same time in the file browser. This feature is not supported by Internet Explorer.
   - Add each file on a separate line by clicking Add Another Attachment, and then clicking Choose Files on the next line. Repeat until all desired files are selected. This feature is available in all supported browsers.
5. Click Attach.

Attached files appear in the Current file attachments list and at the top of the form. A message appears if a file is not attached because it is too large or is a restricted file type. Administrators can configure these limits.
6. Close the pop-up window to return to the form.

Manage attachments

You can view, rename, and remove the attachments on a record, and add additional attachments.

1. Navigate to the record (example, an incident record).

2. Click the attachments icon (🔗) or the Manage Attachments link if files are already attached. The number of attachments is listed on the Manage Attachments link.
3. To rename an attachment, complete the following steps.
   a) Click [rename] beside the file name at the top of the form or in the pop-up window.
      a) Edit the file name and press the Enter key.

      **Note:** Press the Esc key instead of the Enter key to undo your changes and cancel rename.

4. To remove attachments, click the **Manage Attachments** link and complete the following steps.
   a) Select the check boxes next to the attachments to delete.
   b) Click **Remove**.
      The attachment disappears and the pop-up window remains open.

5. To add additional attachments, complete the following steps.
   a) Click **Choose Files**.
   b) Select the file to attach and click **Open**.
   c) Click **Attach**.

6. Close the pop-up window to return to the form.

**Attach files with drag-and-drop**

You can drag files from your computer to a form and to upload them.
This functionality is supported in Firefox 3.6 or later and Chrome. Support will be added for other web browsers as they implement the HTML5 specification.

1. Navigate to the record.
2. On your computer, browse to the files to attach.
3. Select the files in the file browser, and then drag them over the header bar on the form.
4. When the attachments header appears orange, release the mouse button to begin the upload.

When the upload is complete, the file name is added to the attachments header. Upload speed depends on the file size and the speed of your network connection.

**Warning:** Do not navigate away from the record while an upload is in progress. The upload must be completed for the file to be attached.

Checklists

Checklists provide a simple way to track the progress of tasks without creating additional records. Checklists can be added to the form view of any table that extends Task [task].

For example, if a support agent is assigned an incident task to investigate a wireless issue, a checklist can be used to document the individual steps taken.

Checklists are added to records after they are submitted. You cannot add a checklist as you create a record.
Figure 57: Sample checklist

Enabling checklists in forms

To enable checklists, navigate to the form and add the Checklist formatter. For instructions, see Add a formatter on page 714.
Using checklists

You can add, remove, or rearrange checklist items. You can also save a checklist as a template for future use on other records. Any user can create or edit a checklist and check off completed items.

Activate checklists

Checklists are active by default on new instances. For instances upgrading from a previous version, the Checklist plugin must be activated.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Create a new checklist

You can create a unique checklist for each task record.

Have a user with the personalize_form role enable checklists for the form.

Role required: none

1. Navigate to a record that does not already contain a checklist.
2. Click the down arrow beside the Checklist formatter.
3. Select Create new.
4. Click Add Item.
5. Enter text for the checklist item.
6. Press the Enter key to add the checklist item.
7. Create as many additional checklist items as desired.
8. Optional: Click the minus (-) icon to delete a checklist item.
9. Optional: Click the drag icon ( ) and drag a checklist item to a different position in the list.
10. Optional: Save the checklist as a template for easy reuse.
   a) Click the down arrow beside the Checklist formatter.
   b) Select Save as Template.
      A dialog box appears.
   c) Enter a descriptive template name to identify the checklist.
      When a user creates a checklist from a template, all templates are listed in alphabetical order and there is no way to filter which templates appear. To provide a better user experience, consider implementing a naming system for checklist templates. For example, include the name of the table or another identifier to clarify how the checklist should be used.
   d) Click Save.
11. Save the form.

Create a checklist from a template

You can quickly create a checklist from a previously created template.

The Checklist formatter must be added to the form by a user with the personalize_form role.

Role required: none

A template saves time by creating checklist items automatically. You can add, edit, or remove checklist items without impacting the template. You can use any checklist template, even if it was created on a different table.

1. Navigate to a record that does not already contain a checklist.
2. Click the down arrow beside the Checklist formatter.
Figure 58: Checklist with a template selection

Delete a checklist

You can remove a checklist from a record.

Role required: none

1. Navigate to a record that contains a checklist.
2. Click the down arrow beside the **Checklist** formatter.
3. Select **Remove Checklist**.
   A confirmation dialog box appears.
4. Click **Delete**.

Delete a checklist template

You can delete checklist templates you no longer need.

Role required: none

Deleting a checklist template has no effect on checklists created from that template.

1. In the navigation filter, enter `checklist_template.list`.
2. Select the check box by the checklist template you want to delete.
3. In the actions choice list, select **Delete**.
4. In the confirmation dialog box, click **Delete**.

User presence

User presence is a UI16 feature that lets you see who is online when you are working in an instance.

Your avatar appears in the form header next to your name, and in multiple other places such as in activity streams, Visual Task Boards, live feeds, and Connect conversations. Online status is represented by a dot on the user's avatar.

- Green dot if the user is logged in.
- No dot if the user is not logged in.
- Orange dot if the user recently logged out.

The user avatar is the image uploaded to the user record `[sys_user]`. If no image is uploaded, the avatar is the user's initials.

---

**Note:** Users can add an avatar image to their live feed profile. The system uses the live feed avatar if no image is uploaded to the user record.

---

When you are viewing a record in a form, such as an incident, you can see if other users are viewing the same record.
Figure 59: User presence in a form

If multiple users are viewing the record, the avatar is represented by the number of users. Point your cursor to the number to see the names and avatars of the users.

When you are in a Connect conversation or entering comments in an activity stream, you can see information about the other participant’s activity, for example if they are viewing or typing.
An administrator can disable user presence globally.

Disable user presence

You can disable user presence globally by enabling a system property.

Role required: admin

Enabling the property turns off all user presence features. The property is available starting with the Geneva Patch 2 release.

1. Navigate to `sys_properties.list`.
2. Locate the property named `glide.ui.presence.disabled`.
3. Set the Value to true.
Common UI elements

This page lists common UI elements for the standard user interface.

Dot-walking

Dot-walking provides access to fields on related tables from a form, list, or script.

If the current table contains a reference to another table, any field on the referenced table can be accessed using dot-walking.

Dot-walking references a field by building a chain of field names separated by dots (periods). For instance, incident.assigned_to.company references the company of the user assigned to an incident. The recommended limit for chain length is three levels.

Dot-walking in tree pickers

The tree picker interface presents an expandable, hierarchical view that may be used when selecting fields.

For example, this **Users** glide list field has a select fields icon ( ):

![Figure 61: List field](image)

Clicking this icon presents a list of fields available on the current record:
Reference fields have + icons next to their name. Clicking the + expands a list of the fields on that referenced field. In this example, expanding the **Assigned To** field shows the **User** fields for that record:
Selecting the Manager field adds the variable ${assigned_to.manager}:

Dot-walking in choice list fields

Users can dot-walk to related fields in a choice list, such as the field list in a filter. This example demonstrates how to filter the incident table by the company of the caller who registered the incident.

First, open the choice list of fields to filter by. This presents a list of the fields that are present on the Incident [incident] table. Reference fields are followed by related fields, which are presented in blue. For instance: Caller is followed by Caller > User fields. This means that Caller is a reference field, and the related fields are User fields on the Caller record. If the blue related fields are not present in the list, select Show Related Fields in blue at the bottom of the list:
Figure 65: Show related fields

When the related fields are present, select a set of related fields.
Figure 66: Related fields

After the related field is selected, the menu reloads with the related table's fields.
Figure 67: Related table fields

The new field is: `caller.company`.

When looking at the list, it is easy to see where in the dot-walk the user currently is. Each selected reference is stored at the top, and the number of dots in front indicate how many dots from the initial record the user has reached.
In this picture, the user is currently at `incident.caller.company`. It is possible to return to higher levels in the hierarchy by selecting the blue records. For instance, selecting `Incident fields` returns to the list of incident fields.

The related fields can be dismissed by selecting `Remove Related Fields` in blue at the bottom of the list:
Dot-walking in list collectors

When selecting a list of fields from a list collector (for example, when you are configuring a form), it is also possible to dot-walk to fields from other forms.

To see which fields are reference fields and can be dot-walked, look for green fields with a plus symbol.
Figure 70: Reference fields for dot-walking

Once a reference field is highlighted, the expand icon appears above the add icon.
Figure 71: Expand icon

Selecting the expand icon opens the list of fields from the related list in the Available pane. In this picture, the user has opened the Assigned to fields. As in other examples, the previous lists of fields appear at the top of the list in blue.
Figure 72: Lists of fields

Once the field is added to the Selected pane, it appears with its full dot-walked syntax. In the next picture, the user has selected **Assigned_to.active**.
Dot-walking in scripts

You can dot-walk within a script by invoking the dot-walk syntax.

This functionality requires a knowledge of JavaScript.

For scripts that run on the server side, such as business rules, it is necessary to add `current`.

The following script, for example, is a scripted approval rule that requests an approval from the manager of the user who opened the ticket.

```javascript
try {
    current.opened_by.manager;
} catch (err) { }
```

For scripts that run on the client side, such as client scripts, `current` is not necessary. For instance, the following Highlight VIP Caller script runs on the client side.

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    // wait until there is a valid record in the field if (newValue) {
        // get the caller object so we can access fields
        var caller = g_form.getReference('caller_id');
    }
}
```
Geneva  ServiceNow  ServiceNow Platform

```javascript
var callerLabel = document.getElementById('label.incident.caller_id');
var callerField = document.getElementById('sys_display.incident.caller_id');

// check for VIP status if (caller.vip == 'true') {

  // change the caller label to red background //style object
  if (callerLabel)
    document.getElementById('label.incident.caller_id').style.backgroundColor = 'red';

  // change the caller's name field to red text if (callerField)
  if (callerField)
    document.getElementById('sys_display.incident.caller_id').style.color = 'red';

  // not a VIP, remove temporary styles if
  else {
    if (callerLabel)
      document.getElementById('label.incident.caller_id').style.backgroundColor = '';
    if (callerField)
      document.getElementById('sys_display.incident.caller_id').style.color = '';
  }

} }
```

**Dot-walking in variables**

Often, you can add variables into templates, notifications, or other forms where a value is being called from the form.

For example, `{$assigned_to}` is the variable for the **Assigned to** field.

As illustrated above, it is possible to dot-walk to fields on any reference field's original record. In the case above, it is possible to dot-walk to any field on the `assigned_to` record, for example, `{$assigned_to.manager}`.

As is always the case with dot-walking, this can be a longer chain if needed, as in this example: `{$assigned_to.department.manager.mobile_phone}`.

Sometimes this variable can be selected from a tree picker.

**Reference icon**

The reference icon (.circle in UI16 and UI15, in other UIs) appears to the left of a record on a list, or to the right of a reference field on a form populated with a record.

Pointing to the icon presents a read-only pop-up window with the record's information.
Figure 74: Popup window

Note: Users who are using the Windows Live toolbar must disable the toolbar to use reference lookups.

To navigate to the referenced record, click the reference icon. Clicking through a reference icon automatically saves the current record.

To freeze a pop-up window, press the Shift key when pointing to the reference icon, and then move the mouse cursor. The pop-up window remains on the screen until it is closed or another pop-up is displayed.

Note: On some UK keyboards, this behavior is achieved with the Alt key.

Use a slushbucket

Slushbuckets allow you to select multiple items from a list of available items and to remove items from a list of selected items.

Slushbuckets are used in many operations, such as personalizing lists, adding items to related lists, and Service Catalog list collector variables.

The slushbucket interface has two columns: the available items on the left and the selected items on the right.

- To add items to the selection, double-click an available item (on the left), or select an item and click the add icon (>). The new item is added at the bottom of the selected items column on the right.
- To remove items from the selection, double-click the item on the right, or select the item and click the remove icon (<).
- To select multiple consecutive items, hold the Shift key and click the first and last item.
- To select multiple non-consecutive items, hold the Ctrl key (Command key on Mac) and click the desired items.

Arrange slushbucket selections

Some slushbuckets allow users to customize the order of selections, such as when you are configuring a form or list.
Follow one of the two methods below to move an item.

- To move items one position, select the items and click the up or down icon.
- To move an item multiple positions, select consecutive items above or below the item then click the up or down icon. In this example, to move **Assigned to** to the top of the selections order, complete the following steps.
  a) Hold the Shift key and click **Short description** then **Number**.
  b) Click the down icon.

### Personalizing

Find available slushbucket items

Some slushbuckets provide filter and search controls for available items, such as adding items to related lists.

Information fields for the highlighted available item appear beneath some slushbuckets. The information fields are the same as the reference lookup for a reference field on a form. To modify which fields appear, configure the reference lookup list.

- To filter available items, create conditions using the condition builder and click **Run filter**.
- To search available items, enter text in the **Search** field. The list of available items is filtered as you type.
Set the number of slushbucket items visible in the available column

You can modify the glide.xmlhttp.excessive property to change the number of items that appear in the Available column of the slushbucket.

Role required: admin

1. Add the property to the System Properties [sys_properties] table.
2. Change the Type to Integer.
3. Enter a number in the Value field. The default value is 100.
4. Save the property.

Warning: Selecting a number significantly higher than 100 can lead to performance issues on your instance as the data is loaded into the slushbucket.

Tree picker

The tree picker is a special reference lookup for the following items.

• Configuration Items (CIs) for a field that is dependent on another CI field.
• Reference elements for any hierarchical table. A hierarchical table is any table that has a parent field pointing back at itself.
Geneva    ServiceNow    ServiceNow Platform

• Values for a user reference that is dependent on the group.

Add the tree picker attribute

A limit of 1000 has been placed on the number of nodes returned to the tree picker. This limit is configurable with the glide.ui.group heirarchy.max_nodes property.

Role required: personalize_dictionary

1. Modify the dictionary entry for the field.
2. Add tree_picker=true to the Attributes field.
   If there are multiple attributes, use a comma to separate them without any spaces between.

   Attributes: [tree_picker=true]

Tree picker examples

Assignment Group

By default, the Assignment group field uses the tree picker attribute.
This presents a hierarchical tree view of the reference field's options, which in this case is groups. If a group has sub-groups, they appear in the tree structure. It is possible to expand a group to see its members, but members cannot be selected.

**Note:** You cannot customize the label names used in the tree picker. The label names are taken from the values in the table. For example, the Assignment group choices come from the group names in the Name column of the Groups table.

**Simple dependent fields**

It is also possible to add the tree-picker attribute on a reference field that is dependent on a reference field of a different type. For example, if Assigned to is dependent on Assignment group, it is possible to add the tree picker attribute on Assigned to. The result is a one-node tree for the group and its members.
Figure 76: Tree picker on a reference field

**Dependent fields**

In this example, there are two CI reference fields, and one is dependent on the other. The dependent one has the tree_picker attribute:

For this example, the configuration item is Bond Trading. The tree picker for the **Dependent CI** field shows all downstream and upstream relationships for that CI. The small orange square icon above a relationship
indicates that the CI, **lawson**, already has an open task against it. Pointing to the orange square displays a pop-up window with information about the open task.

**Note:** Dependencies for CIs are enforced only when the dictionary entry for the reference field has `tree_picker=true` in the **Attributes** field.

### Condition builder

A condition builder constructs a condition statement with a series of contextually generated fields. Condition builders are used in many operations, such as creating filters, administering surveys, and administering access control.

### Condition builder format

A condition consists of three parts:

- **Field:** a choice list based on the table and user access rights. The choice list can include fields on related tables by dot-walking.
- **Operator:** a choice list based on the field type. For example, in the Incident [incident] table, the `>` operator does not apply to the **Active** field but it does apply to the **Priority** field.
- **Value:** a text entry field or a choice list, depending on field type. For example, in the Incident [incident] table, the **Active** field offers a choice list with the values **true, false, and empty**, while the **Short Description** field offers a text entry field.

### Condition builder actions

You can add a dependent condition by clicking **AND** or **OR** next to the condition. You can add a top-level condition by clicking **AND** or **OR** on the condition builder toolbar above the conditions.

You can remove a condition by clicking the delete icon (X) next to the condition.

![Figure 78: Example AND condition](image-url)

© 2017 ServiceNow. All rights reserved. 157
Filtering on empty fields

Most filter operations do not return empty fields in their result set. You can create a filter that displays records with an empty field value in addition to records that match the initial filter conditions.

For example, when viewing all records that are assigned to the Hardware group, to include records with an empty Assignment group field, complete the following steps.

1. Create the filter condition [Assignment group] [is] [Hardware].
   
   Note: This condition does not return those records where the Assignment group field is empty.

2. Click OR next to the original filter condition.

3. Create an additional filter condition of [Assignment group] [is empty].

4. Run the filter.

Values for date/time fields

When you filter on fields of type date/time, such as the Created field on any task record, several time-related options are available, such as Today, This week, Last 3 months, and so on.

For example, at 12 P.M. on June 1st, a user in New York filters a list of incidents using the Created on Today condition. The resulting list shows all incidents created during the last six hours: between midnight (00:00:00) and the user’s current time, noon (12:00:00), on June 1st. A filter for This week returns incidents created between the previous Monday at midnight to the current day and time.

A filter for Last 3 months returns incidents between midnight on the 1st of the month three full months ago and the current day and time. For example, if you choose the Last 3 months filter on April 15th, the results show records created since January 1st up until April 15. A filter for Last 12 months returns records created from midnight on the first of the month twelve months ago and the current day and time.

For the at or before and at or after filters, use midnight as a start or end point. For example, if you filter a list of incidents created [at or after] [Yesterday], the resulting list shows all incidents created at midnight yesterday or later. If it is Thursday, August 7, and you filter a list of incidents created [at or before] [Last week], the resulting list shows all incidents created at and before midnight on Sunday August 3, which is the end of last week.
Operators available for filters and queries

The system provides a set of operators for use with filters, condition builders, and encoded queries. A field's data type determines what operators are available for it.

### Table 36: Operators available for condition builders

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>[Active][is][True][AND] [Caller][is not empty]</td>
<td>^</td>
<td>active=true*CallerISNOTEMPTY</td>
</tr>
<tr>
<td>OR condition</td>
<td>[Short description][is empty] [OR][Description][is empty]</td>
<td>^OR</td>
<td>short_descriptionISEMPTY^ORdescriptionISEMPTY</td>
</tr>
<tr>
<td>OR filter</td>
<td>All of these conditions must be met [Short description][is empty] OR all of these conditions must be met [Description][is not empty]</td>
<td>^NQ new query</td>
<td>short_descriptionISEMPTY^NQdescriptionISEMPTY</td>
</tr>
</tbody>
</table>

### Table 37: Operators available for string fields

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>starts with</td>
<td>[Short description][starts with][SAP]</td>
<td>STARTSWITH</td>
<td>short_descriptionSTARTSWITHSAP</td>
</tr>
<tr>
<td>ends with</td>
<td>[Short description][ends with][outage]</td>
<td>%</td>
<td>short_descriptionENDSWITHoutage</td>
</tr>
<tr>
<td>contains</td>
<td>[Short description][contains][SAP]</td>
<td>*</td>
<td>short_descriptionLIKESAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LIKE</td>
<td></td>
</tr>
<tr>
<td>does not contain</td>
<td>[Short description][does not contain][SAP]</td>
<td>!*</td>
<td>short_descriptionNOT LIKESAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTLIKE</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td>[Short description][is][Network storage unavailable]</td>
<td>=</td>
<td>short_description=Network storage unavailable</td>
</tr>
<tr>
<td>is not</td>
<td>[Short description][is not][Network storage unavailable]</td>
<td>!=</td>
<td>short_description!=Network storage unavailable</td>
</tr>
<tr>
<td>is empty</td>
<td>[Short description][is empty]</td>
<td>ISEMPTY</td>
<td>short_descriptionISEMPTY</td>
</tr>
<tr>
<td>Operator label</td>
<td>Example condition</td>
<td>Equivalent query operator</td>
<td>Example query</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------</td>
<td>---------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>is not empty</td>
<td>[Short description][is not empty]</td>
<td>ISNOTEMPTY</td>
<td>short_descriptionISNOTEMPTY</td>
</tr>
<tr>
<td>is anything</td>
<td>[Short description][is anything]</td>
<td>ANYTHING</td>
<td>short_descriptionANYTHING</td>
</tr>
<tr>
<td>is empty string</td>
<td>[Short description][is empty string]</td>
<td>EMPTYSTRING</td>
<td>short_descriptionEMPTYSTRING</td>
</tr>
<tr>
<td>less than or is</td>
<td>[Short description][less than or is][s]</td>
<td>&lt;=</td>
<td>short_descriptionlt;=s</td>
</tr>
<tr>
<td>greater than or is</td>
<td>[Short description][greater than or is][s]</td>
<td>&gt;=</td>
<td>short_description&gt;=s</td>
</tr>
<tr>
<td>between</td>
<td>[Short description][between][q] and [t]</td>
<td>BETWEEN</td>
<td>short_descriptionBETWEENq@t</td>
</tr>
<tr>
<td>is same</td>
<td>[Short description][is same] as [Description]</td>
<td>SAMEAS</td>
<td>short_descriptionSAMEASdescription</td>
</tr>
<tr>
<td>is different</td>
<td>[Short description][is different] from [Description]</td>
<td>NSAMEAS</td>
<td>short_descriptionNSAMEASdescription</td>
</tr>
</tbody>
</table>

Table 38: Operators available for reference fields

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td>[Caller][is][Don Goodliffe]</td>
<td>=</td>
<td>caller_id=9ee1b13dc6112271007f9d0efdb69cd0</td>
</tr>
<tr>
<td>is not</td>
<td>[Caller][is not][Don Goodliffe]</td>
<td>!=</td>
<td>caller_id!=9ee1b13dc6112271007f9d0efdb69cd0</td>
</tr>
<tr>
<td>is empty</td>
<td>[Caller][is empty]</td>
<td>IEMPTY</td>
<td>caller_idISEMPTY</td>
</tr>
<tr>
<td>is not empty</td>
<td>[Caller][is not empty]</td>
<td>ISNOTEMPTY</td>
<td>caller_idISNOTEMPTY</td>
</tr>
<tr>
<td>starts with</td>
<td>[Caller][starts with][Don]</td>
<td>STARTSWITH</td>
<td>caller_idSTARTSWITHDon</td>
</tr>
<tr>
<td>ends with</td>
<td>[Caller][ends with][liffe]</td>
<td>%</td>
<td>caller_idENDSWITHliffe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>%</td>
<td>ENDSWITH</td>
</tr>
<tr>
<td>contains</td>
<td>[Caller][contains][on]</td>
<td>•</td>
<td>caller_idLIKEon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>•</td>
<td>LIKE</td>
</tr>
<tr>
<td>does not contain</td>
<td>[Caller][does not contain] [on]</td>
<td>!•</td>
<td>caller_idNOT LIKEon</td>
</tr>
<tr>
<td>is anything</td>
<td>[Caller][is anything]</td>
<td>ANYTHING</td>
<td>caller_idANYTHING</td>
</tr>
<tr>
<td>Operator label</td>
<td>Example condition</td>
<td>Equivalent query operator</td>
<td>Example query</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>is same</td>
<td>[Caller][is same] as [Assigned to]</td>
<td>SAMEAS</td>
<td>caller_idSAMEASassigned_to</td>
</tr>
<tr>
<td>is different</td>
<td>[Caller][is different] from [Assigned to]</td>
<td>NSAMEAS</td>
<td>caller_idNSAMEASassigned_to</td>
</tr>
<tr>
<td>is empty string</td>
<td>[Caller][is empty string]</td>
<td>EMPTYSTRING</td>
<td></td>
</tr>
<tr>
<td>is (dynamic)</td>
<td>[Caller][is (dynamic)][Me]</td>
<td>DYNAMIC</td>
<td>caller_idDYNAMIC54635e965f510100a9ad2572f2b4774</td>
</tr>
</tbody>
</table>

**Note:** Not all operators are available for all reference types. Depending on the reference you select, you may see a shorter list of operators.

---

**Table 39: Operators available for choice fields containing strings**

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td>[Subcategory][is][Email]</td>
<td>=</td>
<td>subcategory=email</td>
</tr>
<tr>
<td>is not</td>
<td>[Subcategory][is not][Email]</td>
<td>!=</td>
<td>subcategory!=email</td>
</tr>
<tr>
<td>is one of</td>
<td>[Subcategory][is one of] [DB2, MS SQL Server, Oracle]</td>
<td>IN</td>
<td>subcategoryINdb2,sql server,oracle</td>
</tr>
<tr>
<td>is not one of</td>
<td>[Subcategory][is not one of][DB2, MS SQL Server, Oracle]</td>
<td>NOT IN</td>
<td>subcategoryNOT INdb2,sql server,oracle</td>
</tr>
<tr>
<td>contains</td>
<td>[Subcategory][contains] [Em]</td>
<td>LIKE</td>
<td>subcategoryLIKEem</td>
</tr>
<tr>
<td>starts with</td>
<td>[Subcategory][starts with] [Em]</td>
<td>STARTSWITH</td>
<td>subcategorySTARTSWITHem</td>
</tr>
<tr>
<td>ends with</td>
<td>[Subcategory][ends with][il]</td>
<td>• %</td>
<td>subcategoryENDSWITHil</td>
</tr>
<tr>
<td>does not contain</td>
<td>[Subcategory][does not contain][Em]</td>
<td>NOT LIKE</td>
<td>subcategoryNOT LIKEem</td>
</tr>
<tr>
<td>is anything</td>
<td>[Subcategory][is anything]</td>
<td>ANYTHING</td>
<td>subcategoryANYTHING</td>
</tr>
<tr>
<td>is same</td>
<td>[Subcategory][is same] as [Category]</td>
<td>SAMEAS</td>
<td>subcategorySAMEAScategory</td>
</tr>
<tr>
<td>is different</td>
<td>[Subcategory][is different] from [Category]</td>
<td>NSAMEAS</td>
<td>subcategoryNSAMEAScategory</td>
</tr>
</tbody>
</table>
### Table 40: Operators available for choice fields containing integers

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td>[Impact][is][1 - High]</td>
<td>=</td>
<td>impact=1</td>
</tr>
<tr>
<td>is not</td>
<td>[Impact][is not][1 - High]</td>
<td>!=</td>
<td>impact!=1</td>
</tr>
<tr>
<td>is one of</td>
<td>[Impact][is one of][1 - High, 2 - Medium]</td>
<td>IN</td>
<td>impactIN1,2</td>
</tr>
<tr>
<td>is not one of</td>
<td>[Impact][is not one of][1 - High, 2 - Medium]</td>
<td>NOT IN</td>
<td>impactNOT IN1,2</td>
</tr>
<tr>
<td>is empty</td>
<td>[Impact][is empty]</td>
<td>EMPTY</td>
<td>impactISEMPTY</td>
</tr>
<tr>
<td>is not empty</td>
<td>[Impact][is not empty]</td>
<td>NOTEMPTY</td>
<td>impactISNOTEMPTY</td>
</tr>
<tr>
<td>less than</td>
<td>[Impact][less than][2 - Medium]</td>
<td>&lt;</td>
<td>impact&lt;2</td>
</tr>
<tr>
<td>greater than</td>
<td>[Impact][greater than][2 - Medium]</td>
<td>&gt;</td>
<td>impact&gt;2</td>
</tr>
<tr>
<td>less than or is</td>
<td>[Impact][less than or is][2 - Medium]</td>
<td>&lt;=</td>
<td>impact&lt;=2</td>
</tr>
<tr>
<td>greater than or is</td>
<td>[Impact][greater than or is][2 - Medium]</td>
<td>&gt;=</td>
<td>impact&gt;=2</td>
</tr>
<tr>
<td>between</td>
<td>[Impact][between][1 - High] and [2 - Medium]</td>
<td>BETWEEN</td>
<td>impactBETWEEN1@2</td>
</tr>
<tr>
<td>is anything</td>
<td>[Impact][is anything]</td>
<td>ANYTHING</td>
<td>impactANYTHING</td>
</tr>
<tr>
<td>is same</td>
<td>[Impact][is same] as [Urgency]</td>
<td>SAMEAS</td>
<td>impactSAMEASurgency</td>
</tr>
<tr>
<td>is different</td>
<td>[Impact][is different] from [Urgency]</td>
<td>NSAMEAS</td>
<td>impactNSAMEASurgency</td>
</tr>
</tbody>
</table>

**Note:** The operators ‘less than or is’ and ‘greater than or is’ are not supported for integer fields in the condition builder. Use a scripted condition instead.

### Table 41: Operators available for date-time fields

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>[SLA due][on][Today]</td>
<td>ONToday</td>
<td>sla_dueONToday@javascript:gs.daysAgoStart(0)@javascript:gs.daysAgoEnd(0)</td>
</tr>
<tr>
<td>not on</td>
<td>[SLA due][not on][Today]</td>
<td>NOTONToday</td>
<td>sla_dueNOTONToday@javascript:gs.daysAgoStart(0)@javascript:gs.daysAgoEnd(0)</td>
</tr>
<tr>
<td>before</td>
<td>[SLA due][before][Today]</td>
<td>&lt;</td>
<td>sla_due&lt;javascript:gs.daysAgoStart(0)</td>
</tr>
<tr>
<td>at or before</td>
<td>[SLA due][at or before][Today]</td>
<td>&lt;=</td>
<td>sla_due&lt;=javascript:gs.daysAgoEnd(0)</td>
</tr>
<tr>
<td>Operator label</td>
<td>Example condition</td>
<td>Equivalent query operator</td>
<td>Example query</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------------</td>
<td>---------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>after</td>
<td>[SLA due][after][Today]</td>
<td>&gt;</td>
<td>sla_due&gt;javascript:gs.daysAgoEnd(0)</td>
</tr>
<tr>
<td>at or after</td>
<td>[SLA due][at or after][Today]</td>
<td>&gt;=</td>
<td>sla_due&gt;=javascript:gs.daysAgoStart(0)</td>
</tr>
<tr>
<td>between</td>
<td>[SLA due][between][Yesterday] and [Today]</td>
<td>BETWEEN</td>
<td>sla_dueBETWEENjavascript:gs.daysAgoStart(1)@javascript:gs.daysAgoEnd(0)</td>
</tr>
<tr>
<td>trend (on or after)</td>
<td>[SLA due][trend][on or after] [Monday]</td>
<td>DATEPART</td>
<td>sla_dueDATEPARTMonday@javascript:gs.datePart('dayofweek','monday','LE')</td>
</tr>
<tr>
<td>trend (on or before)</td>
<td>[SLA due][trend][on or before][Monday]</td>
<td>DATEPART</td>
<td>sla_dueDATEPARTMonday@javascript:gs.datePart('dayofweek','monday','GE')</td>
</tr>
<tr>
<td>trend (after)</td>
<td>[SLA due][trend][after][Monday]</td>
<td>DATEPART</td>
<td>sla_dueDATEPARTMonday@javascript:gs.datePart('dayofweek','monday','GT')</td>
</tr>
<tr>
<td>trend (before)</td>
<td>[SLA due][trend][before][Monday]</td>
<td>DATEPART</td>
<td>sla_dueDATEPARTMonday@javascript:gs.datePart('dayofweek','monday','LT')</td>
</tr>
<tr>
<td>trend (on)</td>
<td>[SLA due][trend][on][Monday]</td>
<td>DATEPART</td>
<td>sla_dueDATEPARTMonday@javascript:gs.datePart('dayofweek','monday','EE')</td>
</tr>
<tr>
<td>relative (on or after)</td>
<td>[SLA due][relative][on or after][1][Hours][ago]</td>
<td>RELATIVEGE</td>
<td>sla_dueRELATIVEGE@hour@ago@1</td>
</tr>
<tr>
<td>relative (on or before)</td>
<td>[SLA due][relative][on or before][1][Hours][ago]</td>
<td>RELATIVELE</td>
<td>sla_dueRELATIVELE@hour@ago@1</td>
</tr>
<tr>
<td>relative (after)</td>
<td>[SLA due][relative][after][1][Hours][ago]</td>
<td>RELATIVEGT</td>
<td>sla_dueRELATIVEGT@hour@ago@1</td>
</tr>
<tr>
<td>relative (before)</td>
<td>[SLA due][relative][before][1][Hours][ago]</td>
<td>RELATIVELT</td>
<td>sla_dueRELATIVELT@hour@ago@1</td>
</tr>
<tr>
<td>relative (on)</td>
<td>[SLA due][relative][on][1][Hours][ago]</td>
<td>RELATIVEEE</td>
<td>sla_dueRELATIVEEE@hour@ago@1</td>
</tr>
<tr>
<td>is empty</td>
<td>[SLA due][is empty]</td>
<td>IEMPTY</td>
<td>sla_dueISEMPTY</td>
</tr>
<tr>
<td>is not empty</td>
<td>[SLA due][is not empty]</td>
<td>ISNOTEMPTY</td>
<td>sla_dueISNOTEMPTY</td>
</tr>
<tr>
<td>is anything</td>
<td>[SLA due][is anything]</td>
<td>ANYTHING</td>
<td>sla_dueANYTHING</td>
</tr>
<tr>
<td>is same</td>
<td>[SLA due][is same] as [Activity due]</td>
<td>SAMEAS</td>
<td>sla_dueSAMEAsactivity_due@day</td>
</tr>
<tr>
<td>is different</td>
<td>[SLA due][is different] from [Activity due]</td>
<td>NSAMEAS</td>
<td>sla_dueNSAMEAsactivity_due@day</td>
</tr>
<tr>
<td>is more than</td>
<td>[SLA due][is more than][1][Days][before][Activity due]</td>
<td>MORETHAN</td>
<td>sla_dueMORETHANactivity_due@daybefore@1</td>
</tr>
<tr>
<td>is less than</td>
<td>[SLA due][is less than][3][Days][before][Activity due]</td>
<td>LESSTHAN</td>
<td>sla_dueLESSTHANactivity_due@daybefore@3</td>
</tr>
<tr>
<td>Operator label</td>
<td>Example condition</td>
<td>Equivalent query operator</td>
<td>Example query</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------</td>
<td>---------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>is</td>
<td>[Reassignment count] [is] [0]</td>
<td>=</td>
<td>reassignment_count=0</td>
</tr>
<tr>
<td>is not</td>
<td>[Reassignment count] [is not] [0]</td>
<td>!=</td>
<td>reassignment_count!=0</td>
</tr>
<tr>
<td>is empty</td>
<td>[Reassignment count] [is empty]</td>
<td>EMPTY</td>
<td>reassignment_countISEMPTY</td>
</tr>
<tr>
<td>is not empty</td>
<td>[Reassignment count] [is not empty]</td>
<td>NOTEMPTY</td>
<td>reassignment_countISNOTEMPTY</td>
</tr>
<tr>
<td>less than</td>
<td>[Reassignment count] [less than] [2]</td>
<td>&lt;</td>
<td>reassignment_count&lt;2</td>
</tr>
<tr>
<td>greater than</td>
<td>[Reassignment count] [greater than] [2]</td>
<td>&gt;</td>
<td>reassignment_count&gt;2</td>
</tr>
<tr>
<td>less than or is</td>
<td>[Reassignment count] [less than or is] [2]</td>
<td>&lt;=</td>
<td>reassignment_count&lt;=2</td>
</tr>
<tr>
<td>greater than or is</td>
<td>[Reassignment count] [greater than or is] [2]</td>
<td>&gt;=</td>
<td>reassignment_count&gt;=2</td>
</tr>
<tr>
<td>between</td>
<td>[Reassignment count] [between] [1] and [2]</td>
<td>BETWEEN</td>
<td>reassignment_countBETWEEN1@2</td>
</tr>
<tr>
<td>is anything</td>
<td>[Reassignment count] [is anything]</td>
<td>ANYTHING</td>
<td>reassignment_countANYTHING</td>
</tr>
<tr>
<td>is same</td>
<td>[Reassignment count] [is same] as [Reopen count]</td>
<td>SAMEAS</td>
<td>reassignment_countSAMEASreopen_count</td>
</tr>
<tr>
<td>is different</td>
<td>[Reassignment count] [is different] from [Reopen count]</td>
<td>NSAMEAS</td>
<td>reassignment_countNSAMEASreopen_count</td>
</tr>
<tr>
<td>greater than field</td>
<td>[Reassignment count] [greater than field] [Reopen count]</td>
<td>GT_FIELD</td>
<td>reassignment_countGT_FIELDreopen_count</td>
</tr>
<tr>
<td>less than field</td>
<td>[Reassignment count] [less than field] [Reopen count]</td>
<td>LT_FIELD</td>
<td>reassignment_countLT_FIELDreopen_count</td>
</tr>
<tr>
<td>greater than or is field</td>
<td>[Reassignment count] [greater than or is field] [Reopen count]</td>
<td>GT_OR_EQUALS_FIELD</td>
<td>reassignment_countGT_OR_EQUALS_FIELDreopen_count</td>
</tr>
<tr>
<td>less than or is field</td>
<td>[Reassignment count] [less than or is field] [Reopen count]</td>
<td>LT_OR_EQUALS_FIELD</td>
<td>reassignment_countLT_OR_EQUALS_FIELDreopen_count</td>
</tr>
</tbody>
</table>
### Table 43: Operators available for boolean fields

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td>[Active][is][true]</td>
<td>=</td>
<td>active=true</td>
</tr>
<tr>
<td>is not</td>
<td>[Active][is not][true]</td>
<td>!=</td>
<td>active!=true</td>
</tr>
<tr>
<td>is empty</td>
<td>[Active][is empty]</td>
<td>IEMPTY</td>
<td>activeIEMPTY</td>
</tr>
<tr>
<td>is not empty</td>
<td>[Active][is not empty]</td>
<td>ISNOTEMPTY</td>
<td>activeISNOTEMPTY</td>
</tr>
<tr>
<td>is anything</td>
<td>[Active][is anything]</td>
<td>ANYTHING</td>
<td>activeANYTHING</td>
</tr>
<tr>
<td>is same</td>
<td>[Active][is same] as [Made SLA]</td>
<td>SAMEAS</td>
<td>activeSAMEASmade_sla</td>
</tr>
<tr>
<td>is different</td>
<td>[Active][is different] from [Made SLA]</td>
<td>NSAMEAS</td>
<td>activeNSAMEASmade_sla</td>
</tr>
</tbody>
</table>

### Table 44: Operators available for email notifications

<table>
<thead>
<tr>
<th>Operator label</th>
<th>Example condition</th>
<th>Equivalent query operator</th>
<th>Example query</th>
</tr>
</thead>
<tbody>
<tr>
<td>changes</td>
<td>[State][changes]</td>
<td>VALCHANGES</td>
<td>StateVALCHANGES</td>
</tr>
<tr>
<td>changes from</td>
<td>[State][changes from] [Awaiting User Info]</td>
<td>CHANGESFROM</td>
<td>MadeSLACHANGESFROM4^EQ</td>
</tr>
<tr>
<td>changes to</td>
<td>[State][changes to] [Awaiting User Info]</td>
<td>CHANGESTO</td>
<td>MadeSLACHANGESTO4^EQ</td>
</tr>
</tbody>
</table>

### Response time indicator

A response time indicator may appear at the bottom right of forms and lists. This indicator provides the processing time, including the total time and the time for each step, for a completed transaction. The following example shows the response time for retrieving a filtered list in a demo instance.

![Response time indicator](image)

**Figure 80: Response time**

The response time text is:

```
Response time(ms): 968, Network: 3, server: 548, browser: 417
```

In this example, the transaction took the following amount of processing time.

- 499 milliseconds total time
- 155 milliseconds on the server
- 172 milliseconds moving data across the network
- 172 milliseconds in the browser, rendering the HTML and parsing and executing JavaScript

Response time appears on most pages. However, it does not appear for simple operations, such as paging through a set of records or changing the sort order of a list, or for the first transaction in a session.

To hide the response time, click the clock ( ). Click the clock again to show the response time.
Point to the clock to view a tool tip with the response time.

To view a detailed breakdown of the browser processing time on forms, click browser.

![Figure 81: Detailed response time information]

Administrators can disable the response time by setting the glide.ui.response_time property to false.

Tags

Tags allow the grouping and organizing of records.
Tags can be visible to any user (global), visible only to specific groups or users, or visible to a single user.

Tag assignment

There are several ways to assign tags to records.

Assign a tag from the list view using inline field editing

You can assign one or more tags to a record directly from the list view.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign an existing tag</td>
<td>Begin typing the tag name, select the tag from the drop-down choices, and press the Enter key.</td>
</tr>
<tr>
<td>Create a new tag</td>
<td>Enter a new tag name and press the Enter key.</td>
</tr>
</tbody>
</table>

The tag is added to the record.
Assign a tag from a list using the action menu

You can assign one or more tags to a record using the action menu in a list.

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to a list.
2. Select the check box beside one or more records.
3. Click **Actions on selected rows** and scroll to the **Assign tag** listing.
4. Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign an existing tag</td>
<td>Select the tag listed.</td>
</tr>
<tr>
<td>Create a new tag</td>
<td>1. Select <strong>New</strong>. 2. Enter a new tag name and select the appropriate sharing level. 3. Click <strong>Save</strong>.</td>
</tr>
</tbody>
</table>

The tag is added to the record.

Assign a tag from the list context menu

You can assign one or more tags from the list context menu.

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to a list.
2. Right-click a record.
3. Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign an existing tag</td>
<td>Select <strong>Assign Tag &gt; [Tag Name]</strong>.</td>
</tr>
<tr>
<td>Create a new tag</td>
<td>1. Select <strong>Assign Tag &gt; New Tag</strong>. 2. Enter a new tag name and select the appropriate sharing level. 3. Click <strong>Save</strong>.</td>
</tr>
</tbody>
</table>

Assign a tag using the edit tags icon

You can assign one or more tags to a record using the edit tags icon in the form header.

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to a form.
2. Perform the appropriate action for your version of the UI.
1. Click the more options icon (•••) in the form header.
2. Select Add Tag.

3. Click in the Add Tag field.
4. Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign an existing tag</td>
<td>Begin typing the tag name and select the tag.</td>
</tr>
<tr>
<td>Create a new tag</td>
<td>Enter a new tag name and press the Enter key.</td>
</tr>
</tbody>
</table>

The tag is added to the record.

**Configure tags to assign automatically**

You can configure the system to automatically assign a tag to records that match conditions defined in the tag record.

Role required: tags_admin

For example, you can create tags that group high urgency incidents, overdue incidents, canceled changes, or any other set of records. The system automatically assigns the tag to records that match the criteria and removes the tag from records that no longer match the criteria.

1. Navigate to System Definition > Tags.
2. Open the tag that you want to assign automatically.
3. Configure the form to add the Conditions for Tags embedded list.
4. Create a new row in the Conditions for Tags list.
5. Click the reference icon in the Table column.
   The Label Table list opens in a pop-up window.
6. Click New.
7. Enter a Name, select a Table, and add conditions for the automatic assignment of the tag.
8. Click Submit.
9. Click the check mark icon to save the new row.
10. Click Update.

**Edit tags from the list view**

In the list view, you can edit tags that you created.

Personalize the list to display the Tags column.

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to a list that contains records you have previously tagged.
2. Click the edit tag audience icon (•••) beside the tag name.
   The Tag Details dialog box opens.
3. Edit the **Name** field as necessary.
4. Select an option for the **Viewable by** field to set the sharing level.
5. Click **Save**.
   All records associated with the tag are updated to reflect the change.

---

## Edit tags from the form view

From the form view, you can edit tags that you created.

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to a record you have tagged.
2. Perform the appropriate action for your version of the UI.

| UI16 | 1. Click the more options icon (●) in the form header.  
|      | 2. Click the edit tag audience icon (●) beside the tag name. |
| UI15 | 1. Click the edit tags icon (●).  
|      | 2. Click the edit tag audience icon (●) beside the tag name. |

The Tag Details dialog box opens.

3. Edit the **Name** field as necessary.
4. Edit the **Viewable by** field to set the sharing level.
5. Click **Save**.
   All records associated with the tag are updated to reflect the change.

---

## Edit tags from the My Tags module

From the My Tags module, you can edit tags you created.

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to **Self-Service > My Tags**.
2. Open a tag.
3. Make the necessary changes.
4. Click **Update**.
   All records associated with the tag are updated to reflect the change.
Edit tags from the Tags module

From the Tags module, you can edit all tags.
Role required: tags_admin

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to **System Definition > Tags**.
2. Open a tag.
3. Make the necessary changes.
4. Click **Update**.

Edit tags from the Tagged Documents page

You can edit tags from the Tagged Documents page.

**Note:** Tag names are case-insensitive and must not include punctuation.

1. Navigate to **Self-Service > My Tagged Documents**.
   Alternatively, in UI15 you can click **Tagged Documents** in the Edge.
2. Click **Edit Tags**.
   A list of tags you have created appears.
3. Open a tag.
4. Make the necessary changes.
5. Click **Update**.
   All records associated with the tag are updated to reflect the change.

Tag sharing levels

You can set the tag sharing level so other users can use the tag.

The **Viewable by** field that appears when you edit a tag controls the sharing level. The following sharing levels are available.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Me</td>
<td>Private tag, visible only to the owner.</td>
</tr>
<tr>
<td>Groups and Users</td>
<td>Shared tag, visible to the owner and specific groups or users.</td>
</tr>
<tr>
<td>Everyone</td>
<td>Shared tag, visible to everyone. This option is available to users with the admin or tags_admin role.</td>
</tr>
</tbody>
</table>

Tag name conflict

Conflicts can occur if multiple tags have the same name.
If a private tag and a shared tag have the same name, the system appends [private] to the name of the private tag. For example, Sample tag [private]. For definitions of private and shared tags, see Tag sharing levels on page 170.

There cannot be more than one shared tag with the same name. If a user attempts to create a shared tag with a name that is already in use, an error message appears and the system prevents the tag from being saved.

Filter records by tag

You can filter records by tags you have access to.

Filter records by tag in any of the following ways.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter from the form view</td>
<td>Perform the appropriate action for your version of the UI.</td>
</tr>
<tr>
<td></td>
<td>• UI16: Click the more options icon ( ), and then click the tag in the menu that opens.</td>
</tr>
<tr>
<td></td>
<td>• UI15: Click the edit tags icon ( ), and then click the tag.</td>
</tr>
<tr>
<td>Filter from the list view using the Tags column</td>
<td>With the Tags column visible, click the tag name. Clicking two or more tags filters the tags together with the AND operator and shows records containing all of the selected tags.</td>
</tr>
<tr>
<td>Filter from the list view using the list filter</td>
<td>Click the show/hide filter icon ( ), select Tags from the Choose field choice list, and select or enter the tag name.</td>
</tr>
</tbody>
</table>

View tagged documents

The Tagged Documents page displays recently viewed documents or user-tagged documents in the content frame.

Role required: none

An administrator can configure the fields that appear on the tagged document cards by configuring the mobile view of the task, for example, the mobile view of the Incident form. If there is not a mobile view available for the page, the record reverts to the default view. For more information, see Create a view on page 237.

A sidebar displays all the tags you have access to. A preview area displays tagged documents.

2. Optional: Click **Show Global Tags** to display global tags.

3. Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display most recently viewed documents</td>
<td>Click <strong>Most Recent</strong> in the sidebar.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Display documents associated with a tag</td>
<td>Click the tag name in the sidebar.</td>
</tr>
<tr>
<td>Remove a tag from a record</td>
<td>Click the X in the corner of the document preview.</td>
</tr>
<tr>
<td>Open the form for a tagged document</td>
<td>Click the title of the document preview.</td>
</tr>
</tbody>
</table>

### Remove a tag from a record

There are many ways to remove a tag from a record.

Perform any of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove a tag from a record in the list view using the Tags column</td>
<td>Ensure that the Tags column is visible and click the remove tag icon (X) beside the tag name.</td>
</tr>
<tr>
<td>Remove a tag from one or more records in the list view using the action menu</td>
<td>Select the check box for one or more records, then click <strong>Actions on selected rows &gt; Remove Tag: [Tag name]</strong>.</td>
</tr>
<tr>
<td>Remove a tag from a record in the form view</td>
<td>Perform the appropriate action for your version of the UI.</td>
</tr>
<tr>
<td></td>
<td>• UI16: Click the more options icon (⋮) in the form header, and then click the remove tag icon (X).</td>
</tr>
<tr>
<td></td>
<td>• UI15: Click the edit tags icon (ставил), and then click the remove tag icon (X).</td>
</tr>
<tr>
<td>Remove a tag from a record in the Tagged Documents page</td>
<td>Click the remove tag icon in the corner of the document preview.</td>
</tr>
</tbody>
</table>

If a tag is automatically applied to a record based on specific conditions, the tag is automatically removed when those conditions no longer apply. For more information, see *Configure tags to assign automatically* on page 168.

### Merge tags

You can merge one or more tags with another tag. For example, when you merge Tag A with Tag B, Tag A is deleted and all associated records are reassigned Tag B.

Role required: A user with the admin or tags_admin role can merge any tag into any other tag. All other users can merge tags they created into tags that are visible to them.

1. Navigate to **Self-Service > My Tags**.
2. Select the check box for each tag you want to merge with another tag.
3. Select **Actions on selected rows > Merge Tags**.
4. Enter the tag to merge other tags with.
5. Click **OK**.
Create a tag from the Tags list

You can create a tag directly from the Tags list.

Role required: tags_admin

1. Navigate to **System Definition > Tags** or **Self-Service > My Tags**.
2. Click **New**.
3. Enter a name for the tag in the **Name** field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Specifies the owner of the tag. Users are allowed to change the ownership of the tag to someone else.</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of tag.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Standard</strong>: is controlled by users who have access to the tag. These users can add or remove records from the tag.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Most Active</strong>: displays the most frequently accessed modules and is automatically maintained by the system. There is no maximum duration a module can remain as most active.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Most Recent</strong>: displays the most recently accessed modules and is automatically maintained by the system.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Most Active Record</strong>: displays the most frequently viewed records and is automatically maintained by the system. There is no maximum duration a record can remain as most active.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Most Recent Record</strong>: displays the most recently viewed records and is automatically maintained by the system.</td>
</tr>
</tbody>
</table>

Only tags of the **Standard** or **Most Recent Record** types appear on the Tagged Documents page. You can view other types of tags by adding them to a homepage. For more information, see **Add a tag to a homepage** on page 175.

<table>
<thead>
<tr>
<th>Viewable by</th>
<th>Specifies the sharing level. By default tags are visible only to the user (<strong>Me</strong>), but you can share tags with <strong>Groups and Users</strong>. Users with the tags_admin role are the only users who can create global tags shared with <strong>Everyone</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Specifies whether the tag is enabled or disabled.</td>
</tr>
</tbody>
</table>

Configure Zing tag indexing for text search

Zing text indexing is available for tags on records.
Role required: admin

No tags are indexed by default. You can enable text indexing of tags on a table-by-table basis. Private tags are never indexed.

1. Navigate to the dictionary entry for the table you want to turn on indexing for. For more information, see Modify Dictionary Entries from the Dictionary module on page 1445.
2. In the Attribute related list, click New.
3. Click the reference icon to see available attributes.
4. Select the Text Index Tags attribute (text_index_tags).
5. For the attribute value, enter one of the following options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>everyone_only</td>
<td>Only tags shared with everyone will be indexed.</td>
</tr>
<tr>
<td>all_shared</td>
<td>All shared tags (Everyone, Groups and Users) will be indexed.</td>
</tr>
</tbody>
</table>

6. Click Submit.
7. Under Related Links, click Generate Text Index to reindex the table.

After you turn on text indexing for tags on records, records are reindexed when any of the following actions occurs.
• When you add or remove tags.
• When you modify a tag, provided that fewer than 100 records are affected. If there are more records, you must manually reindex.
• When you manually reindex an entire table (required after initial tag indexing setup).

Configure notifications for tagged records

You can enable the system to send a notification when a record with a certain tag is updated.

Role required: tags_admin

1. Navigate to System Definition > Tags.
2. Open a tag.
3. In the Label Entries related list, open the record for which you want notifications.
4. Select the Notify onchange check box to be notified any time the record is modified.
5. Use the Notify when condition builder to specify the conditions that must be true to trigger the notification.
6. Select a business rule in the Notify script field.
7. Save the record.

Add a tag to a homepage

You can add a tag to a homepage to view records to which the tag has recently been assigned.

You might want to do this for tags of a Type other than Standard or Most Recent Record, as they do not appear on the Tagged Documents page.

1. Open a homepage to which you have permission to add content.
2. Click the add content icon (+).
3. Select Labels in the left column of the Add content window.
4. Select the tag you want to show on the homepage.
5. Click the appropriate Add here button.
6. Close the Add content window.

History of recently-viewed records

In UI11, the system offers a viewable history of an ITIL user's recently-viewed records, and it can display an ITIL user's most viewed items.

The system offers a viewable history of an ITIL user's recently-viewed records, and it can display an ITIL user's most viewed items. Both are implemented as tags, and once activated they will appear in the application navigator in UI11.

Figure 82: View history

Note: In UI16, a similar feature is available. For more information, see View your navigation history on page 37.

Activate Most Active and Most Recent tags

To activate Most Active and Most Recent tags in UI11, you must first have viewed some records within the platform.

Role required: itil

After this has happened, perform the following steps to activate Most Active and Most Recent tags.

1. Navigate to your homepage.
2. Click Add content.
3. Select Labels from the first pane.
4. Select Most Recent from the second pane to activate that tag. It is not necessary to click Add.
5. Select Most Active from the second pane to activate that tag. It is not necessary to click Add.
6. After selecting the tags you want to activate, refresh your browser (F5).
The tags should now be visible in the application navigator. Items in the Most Recent tag are listed chronologically in reverse order, as you would expect from a history list.

Deactivate Most Active and Most Recent tags

If you do not wish to see the Most Active and Most Recent tags after they have activated them, they can be deactivated.

Role required: itil

1. Click Most Active or Most Recent in the application navigator.
   The Tag form appears.
2. Clear the Active check box.
3. Click Update.
   Upon browser refresh, the tag will no longer be displayed.

Remove the Most Active and Most Recent tags

Role required: tags_admin

1. From the left navigation pane, select System Definition > Tags.
2. Delete everything with a name of Most Active or Most Recent.
3. From the left navigation pane, select System UI > Widgets, and deactivate the Labels widget.
   This prevents it from showing up in the Add Content part of homepages.

User interface configuration

Change appearance, navigation menus, CSS, and utilize cutting edge interface tools.

Navigation stack

The navigation stack is the portion of the ServiceNow suite of applications that determines where a user will be redirected after the update of a record.

It is also possible to cause page references to be manually inserted into the the Navigation Stack when a link in the Navigation Page is clicked. This is done by modifying a module definition to include an


argument of $sysparm_stack$ with a string value equal to the URL reference to the page that should be inserted into the stack. Finally it is also possible to override the redirection determined by the stack entirely by creating a business rule or editing an existing one to include gs.setRedirect("http://redirect_page.com").

### Operating parameters of the Navigation Stack

The Navigation Stack can be thought of a user’s navigation history in the ServiceNow system. It is updated every time a user views a new page. The Navigation Stack is referenced when users press the ServiceNow back button. It is also referenced when a user updates a record, at which point the user is redirected to their last page in the Navigation Stack.

![Figure 83: Back and Update](image)

**Inserting pages to the Navigation Stack**

It is also possible to cause page references to be manually inserted into the Navigation Stack when a link in the Navigation Page is clicked. This is done by modifying a module definition to include an argument of $sysparm_stack$ with a string value equal to the URL reference to the page that should be inserted into the stack. The following screen shot shows the out of the box definition for the Create New Incident Module. By passing an argument of "sysparm_stack=incident_list.do" we are adding the page "incident_list.do" (the list view of incidents) to the stack so that after creating a new incident this will be the page that a user is redirected to.
Overriding the redirection behaviors of the Navigation Stack with onEvent business rules

The redirection behaviors that occur as a result of Navigation Stack behaviors can be overridden using onEvent business rules. This is done using the Glide System function `gs.setRedirect`. Doing so will override the redirection as determined by the **Navigation Stack**, when the particular even specified occurs. The function `gs.setRedirect` takes a string argument that is a URL for the page that it is desired that the user be redirected to. This can be an external URL such as http://www.Google.com.
The Request Manager allows users to cancel any navigation action they select from the application navigator by clicking another application navigator link. This allows users to cancel slow loading transactions without having to wait for the previous transaction to complete.

One of the following conditions must be true for the Request Manager to cancel a transaction.

- Both the original transaction and new transaction are cancelable (both transactions have the setting `sysparm_cancelable=true`).
- The original transaction is a member of the always cancelable white list (the module is listed in the `glide.request_manager.always_cancel` system property) and the new transaction is cancelable (has the setting `sysparm_cancelable=true`).
- Both the original and new transaction produce an auto-completer query in a reference field.
Cancellation workflow

The Request Manager uses the following workflow to determine whether to cancel a transaction.
Add sites to the always cancel white list

You can add UI pages or other links to the always cancel white list so that users can cancel actions on these pages by navigating to another module or link.

Role required: admin

Items on the white list can always be canceled by user actions. It is recommended that you only add items to the white list that provide read-only data to the user, such as a homepage or a report.

1. Enter `sys_properties.list` in the navigation filter and press the Enter key.
2. In the Go to field, select Name, enter `glide.request_manager.always_cancel`, and click Go.
3. Click the property name to open it.
4. In the Value field, add a comma-separated list of URIs you want to allow users to cancel.
5. Click Update.

Prevent users from canceling module transactions

You can explicitly prevent users from canceling a module’s activity by updating the module definition.

Role required: admin

1. Perform the appropriate action for your version of the UI:

| UI16       | 1. Navigate to System Definition > Application Menus.  
|           | 2. Open the application menu to which you want to add the survey module. |
| UI15 or UI11 | Right-click the application menu you want to add the module to and select Edit Application Menu. |

2. In the Modules related list, click the module you want to prevent users from canceling.
3. Configure the form layout and add the field Uncancellable by Other Modules.
4. Select the check box for Uncancellable by Other Modules.

5. Click Update.
Navigation cancellation system properties

Administrators can control automatic cancellation behaviors with the following system properties.

**Table 46: System Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.request_manager.cancel_other_transaction | Enables or disables automatic cancellation behaviors. When true, users can cancel one navigation action with another navigation action.  
  **Type:** true | false  
  **Default value:** true  
  **Location:** sys_properties table |
| glide.request_manager.always_cancel           | A comma-separated list of URIs that users can always cancel by clicking another navigation action. Typically items on this list are modules that provide read-only data such as homepages, reports, or knowledge articles.  
  **Type:** string  
  **Default value:** home,sys_report_template  
  **Location:** sys_properties table |
| glide.request_manager.cancel_reference_complete | Cancels a user query in a reference field when the user enters more information. For example, if the user starts an AJAX search in the Problem number field by entering PRB and then enters PRB000, the second query cancels the first query.  
  **Type:** true | false  
  **Default value:** true  
  **Location:** sys_properties table |

**Use cases**

There are three use cases where administrators can control user cancellation actions.

1. A user clicks on a module, such as the homepage (`home.do`), and while waiting for the module to load decides to navigate to another module, such as the list of open incidents (`incident_list.do`). The system property `glide.request_manager.cancel_other_transaction` controls this use case.

2. A user clicks on a module that displays read-only data, such as a report (`sys_report_template.do?sysparm=sysid`), and while waiting for the module to load decides to navigate to another module, such as the list of open incidents (`incident_list.do`). The system properties `glide.request_manager.cancel_other_transaction` and `glide.request_manager.always_cancel` control this use case.

3. A user starts typing a query in a reference field, such as entering PRB in the related Problem field and while waiting for the auto-completer to display results, decides to enter more information,
such as PRB000. The system properties glide.request_manager.cancel_other_transaction and glide.request_manager.cancel_reference_completer control this use case.

Context-sensitive help

By default, the help link available in the product opens the welcome page of the help system, allowing you to enter search criteria for the information you want to find. The ServiceNow system provides a number of preconfigured help contexts connected to this link that display the exact information you need for the current list, form, or record. You can define your own help topics and configure them to appear when you click the help icon in a ServiceNow record.

Watch Introducing Context-Sensitive Help:

Access context-sensitive help

The location of the help icon depends on the user interface version you are using.

**UI 15**

Click the gear icon in the banner frame to open the menu containing the help icon. The appearance of the menu and icons might be different, depending on the UI scheme you have configured for your instance.

![Context help access in UI 15](image)

**Figure 88: Context help access in UI 15**

**UI 16**

Click the help icon in the right corner of the header bar and select **Search Product Documentation** from the menu.
Administer context-sensitive help

Context-sensitive help can be used in its default configuration with no customizations.

Users click the help icon to open the default help pages provided in the base system. For any page that does not have context-sensitive help defined, the instance displays the help system welcome page. Users can use the search feature or the index to find the correct help topic.

Alternatively, administrators can create custom context-sensitive help to suit the needs of their organizations. Create a new help context that links the help icon to a topic describing a list, form, or specific record. You may initially want to set a base URL to direct to a help system other than the default ServiceNow help system.

This video demonstrates how to set up custom help.

Create a new help context

You can create new help contexts to supplement or replace the default help contexts. For example, if your organization has heavily customized a form, you might create a new help context for that form. The customized help context could link to more relevant information, such as a company knowledge base article.

1. Navigate to System UI > Help Contexts.
2. Click New.
3. Complete the Help Context form from the fields in the table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Select Form, List, or Record. If you select Record, the Table name field is replaced by the Document field.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Table name</td>
<td>Select the table for which the help context is being defined. If the <strong>Type</strong> is <strong>Record</strong>, this field is replaced by the <strong>Document</strong> field.</td>
</tr>
<tr>
<td>Document</td>
<td>Click the reference lookup icon (🔍) to open the document selection dialog box (pictured). Select the table and document (record) for this help context. This field is only visible if you have selected <strong>Record</strong> as the <strong>Type</strong>.</td>
</tr>
<tr>
<td>Language</td>
<td>Select the language of the help page to which you are linking. The choices available depend on the <strong>internationalization plugins</strong> active on your instance. This field allows you to create multiple help contexts for the same form, list, or record, each directing to a help page in a different language. For example, you might have two help contexts for Form X: one for English and the other for Spanish. If a user whose language is set to Spanish clicks the help icon while viewing Form X, the help page defined in the Spanish help context opens.</td>
</tr>
</tbody>
</table>
Geneva    ServiceNow    ServiceNow Platform

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Select this check box to have the system use the help context. This check box is selected by default for new help contexts.</td>
</tr>
<tr>
<td>ServiceNow Wiki</td>
<td>Indicates if this help was created by a customer or was provided by ServiceNow. Clear this check box to create custom help for your organization. Do not attempt to modify existing help topics provided by ServiceNow for your own use. The system is configured to use your custom records rather than the default help contexts, even when they are created on the same table.</td>
</tr>
<tr>
<td>URL or page name</td>
<td>Identifies the topic to display when a user clicks the help icon while viewing a form, list, or record. You can enter the complete URL of the help page or just the target file name if you set a base URL in the help.base.default system property.</td>
</tr>
</tbody>
</table>

**Note:** If you reconfigure the fields on the form, you see the available fields Plugin ID and System Property Base URL. Do not use these fields. They are for internal use only.

4. Click **Submit**.

**Context-sensitive help properties**

The context-sensitive system properties define the base URLs for the default and custom help systems. To access the context-sensitive help properties, navigate to **sys_properties.list** and filter by **Name**. You can edit a property value from the list or by opening the property record.

**Table 48: Context-sensitive help properties**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.help.default.page</td>
<td>The default URL the help icon directs to when no help context is specified. This is the base URL of the ServiceNow help system and opens to the welcome page. Do not change this value.</td>
<td><a href="http://geneva-docs.servicenow.com/?context=">http://geneva-docs.servicenow.com/?context=</a></td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>help.base.default</td>
<td>The base URL for custom help contexts in which the ServiceNow Wiki check box is cleared, and an absolute URL is not specified. When the base URL is set, the system creates the address of a help context record by appending the target page or file name to the base URL. Custom help contexts using this base URL overwrite the help provided by default in your instance for the same table. For details, see Set Base URLs.</td>
<td><a href="http://geneva-docs.servicenow.com/?context=">http://geneva-docs.servicenow.com/?context=</a></td>
</tr>
<tr>
<td>help.base.servicenow</td>
<td>The base URL for help contexts in which the ServiceNow Wiki check box is selected. This is the base URL for the help provided in your instance by default. Do not change this value. For details, see Create a new help context on page 187.</td>
<td><a href="http://geneva-docs.servicenow.com/?context=">http://geneva-docs.servicenow.com/?context=</a></td>
</tr>
</tbody>
</table>

Set base URLs

With context-sensitive help, you can set a base URL for an external help system.

If you want to create numerous help contexts that direct to a single server other than the ServiceNow help system, enter the base URL in the help.base.default property.
Figure 90: Base URL for custom help contexts

On the Help Context form, when the ServiceNow Wiki check box is cleared and you enter a value in the URL or page name field that does not contain the string ://, that value is appended to the base URL defined by help.base.default property.

**Note:** When the ServiceNow Wiki check box is cleared and the URL or page name field contains an absolute URL (distinguished by the string :// in the value), the value in the help.base.default property is ignored. This allows you to create help contexts that link to a variety of different servers.

Help context types

You can create a help context with the type Form, List, or Record, and link it to a particular table in your ServiceNow instance.

When a user clicks the help icon, the system analyzes the relevant help contexts to determine which help page to display.

- Record-level help applies to only one specific record, not the list or form for that table.
For example, if a Record type help context exists for the Validate Number record in the Business Rule table, the help icon only directs to the page specified when a user views that record.

- List-level help applies to the list for a table. If no form-level help is defined, list-level help also applies to the form for the same table.

For example, if a List type but not a Form type help context exists for the Business Rule table, the help icon directs to the page specified by the List type help context when a user views any list or record (if record-level help is not defined for it) in the Business Rule table.

- Form-level help applies to the form for a table. If no list-level help is defined, the form-level help also applies to the list for the same table.

For example, if a Form type but not a List type help context exists for the Business Rule table, the help icon directs to the page specified by the Form type help context when you view any record (if record-level help is not defined for it) in the Business Rule table.

- If both list- and form-level help are defined for a table, the appropriate help is displayed for the list and the form.

For example, if a List type and a Form type help context exist for the Business Rule table, the help icon directs to the page specified by the List type help context when a user views the Business Rules list. The help icon directs to the page specified by the Form type help context when a user views any record (if record-level help is not defined for it) in the Business Rule table.

Help context prioritization

Sometimes multiple help contexts may apply to what a user views.

When there are several matching help context records, the following rules determine which help context is used, in descending order of priority.

1. A customer-created help context is used instead of a default help context provided in the instance for the same table.

2. A help context that matches the user's language setting is used instead of a help context in the default language of the instance. See the Language field on the Help Context form.

3. A help context for a closer table in an extended table hierarchy is used instead of a help context for a further table.

Consider the case of the Linux Server [cmdb_ci_linux_server] table, which has the following parentage: cmdb_ci > cmdb_ci_hardware > cmdb_ci_computer > cmdb_ci_server > cmdb_ci_linux_server. If help contexts exist for both the cmdb_ci_server table and the cmdb_ci table, the help icon directs to the page specified by the cmdb_ci_server help context when you view a record in the cmdb_ci_linux_server table.

System user guide

The system user guide enables you to create end user help documentation that is specific to the policies and procedures of your organization. A default help page is provided in the base system that displays UI16 help documents for system navigation and other basic operations.

The user guide provides the tools for designing help portal pages containing feature-specific help documents. You can create custom help pages and deploy them in various ways using controls in the system. User guide documents are grouped in relevant sections on a help page, using a two column format. Configuration allows you to display the sections and the documents within them in any order. You can display a section on more than one help page. The user guide is supported in UI16, UI15, and UI11.
A default help page is provided in the base system that displays help documents for system navigation and other basic operations. This page is accessible from the help icon in the header bar for UI16 users. To make this page available to users of other supported UI versions, you must create a *programmatic entry point*.

![Figure 91: Basic user guide help page](image)
Activate the system user guide on an upgraded instance

The User Guide (com.glide.user_guide) plugin is active by default for all new instances, but must be activated by a system administrator on upgraded instances.

Role required: admin

Before you activate the User Guide plugin on an upgraded instance, you must also activate the UI16 (com.glide.ui.ui16) plugin. Use this procedure to activate each plugin.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Tables installed with the system user guide

These tables are installed with the User Guide plugin (com.glide.user_guide).

These tables extend the Application File [sys_metadata] table.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Page [sys_product_help_page]</td>
<td>Stores all the help documents and their section assignments.</td>
</tr>
<tr>
<td>Help Section [sys_product_help_section]</td>
<td>Stores all the help sections for the user guide.</td>
</tr>
<tr>
<td>Help [sys_product_help]</td>
<td>Identifies a help page and provides the entry point for linking to it.</td>
</tr>
<tr>
<td>Page to Section [sys_product_help_p2s]</td>
<td>Stores the relationship of the sections in a page to one another.</td>
</tr>
</tbody>
</table>

Create a user guide help page

A user guide help page is a portal displayed to the end user that contains links to related help documents.

Role required: admin

2. Complete the form.
Table 50: Help page field definitions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Page identification used by the system to create a help URL.</td>
</tr>
<tr>
<td>Application</td>
<td>Application scope for this help page. This field is read-only and defaults to the Global scope.</td>
</tr>
<tr>
<td>Title</td>
<td>Display title for this page.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

Create a user guide section

The sections in a system user guide page are the category containers for the individual help documents.

Role required: admin

Create your help page sections separately and then associate them to the page you created.

1. Navigate to **System User Guide > Administration > Help Page Sections** and click **New**.
2. Give your section a unique and concise title, and then click **Submit**. The view returns to the list of sections.
3. Create additional sections for your help page.

### Create user guide help documents

User guide help documents are grouped in appropriate sections on the page and listed in a configurable order.

Create all your help documents from within the section record. This allows you to refine their titles, adjust their placement, and decide if they are appropriate for the section.

Role required: **admin**

1. Navigate to **System User Guide > Administration > Help Page Sections**.
2. Open a section record you created.
3. In the **Help** related list, click **New** to create a help document for that section.
4. Complete the form.

### Table 51: Help document field definitions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Document number generated automatically by the system.</td>
</tr>
<tr>
<td>Section</td>
<td>Section in which this document appears.</td>
</tr>
<tr>
<td>Order</td>
<td>Listing order for this document in the selected section.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Published</td>
<td>Date this document was created.</td>
</tr>
<tr>
<td>Updated</td>
<td>Date this document was last updated.</td>
</tr>
<tr>
<td>Short description</td>
<td>Brief description of the content of this document. This description is used as the title for the document on the help page.</td>
</tr>
<tr>
<td>Text</td>
<td>Document content, including graphics and tables. You can format your entry, create bulleted or numbered lists, and attach images.</td>
</tr>
</tbody>
</table>
To view your queue and take a survey:

1. Navigate to Self-Service > My Assessments & Surveys.
2. Click Take Survey on a survey card to open the questionnaire.

My Assessments and Surveys

Groups
- Assigned to: [name]
- State: [progress]
- Due Date: [date]

Vendor
- Assigned to: [name]
- State: [progress]
- Due Date: [date]

Service Desk Satisfaction
- Assigned to: [name]
- State: [progress]
- Due Date: [date]
5. Click **Submit**.  
The view returns to the section record.

6. Repeat the process to create the necessary help documents for that section.

---

### Configure the help page

After creating the sections and help documents for your user guide, arrange the sections on the help page.

**Role required:** admin

1. Navigate to **System User Guide > Administration > Help Pages** and open your new page.

2. In the **Page to Sections** related list, click **New**.

3. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>Column location on the page for this section. The first column is designated as 0 and the second column is designated as 1. There must be at least one entry for column 0 for the sections to display.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>Order in which the sections display in the columns on the page. If no order number is defined, the system lists the sections in alphabetical order.</td>
</tr>
<tr>
<td>Application</td>
<td>Application scope for this record. This field is read-only and defaults to the <strong>Global</strong> scope.</td>
</tr>
<tr>
<td>Page</td>
<td>Help page on which this section appears. You can use the lookup field to change the page on which a section appears.</td>
</tr>
<tr>
<td>Section</td>
<td>Name of the section to display on the selected page.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.
5. The view returns to the Help Page form.
6. Continue adding the sections to the help page.

**User guide page entry points**

You can create entry points in your instance for help generated with the system user guide by providing the correct address parameters in the target URL.

Help pages generated with the user guide can be displayed to end users from configurable entry points in the system, including **application modules** and **UI actions**. To create these entry points, you must express a target URL using these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Provides</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h.do</td>
<td>Processor that renders the banner frame at the top of the page.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Provides</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$sys_product_help.do</td>
<td>Displays a page containing links to all the help in the user guide. This includes the basic help system and any custom help pages you have created. An example URL would be <a href="https://myinstance.service-now.com/$sys_product_help.do">https://myinstance.service-now.com/$sys_product_help.do</a>.</td>
</tr>
<tr>
<td>$h.do?sysparm_id=basics</td>
<td>Displays the basic help page included with the system. To create a target for a custom user guide, replace id=basic with the ID for your custom page, such as id=surveys. An example of this would be <a href="https://myinstance.service-now.com/$h.do?sysparm_id=surveys">https://myinstance.service-now.com/$h.do?sysparm_id=surveys</a>.</td>
</tr>
</tbody>
</table>

Creating a change password module

To allow users to change their own password, you can create a simple module in Self-Service.

**Prerequisites**

Role required: admin

Create a module in the Self-Service application called Change Password, give it a **Type** of **URL**, and give it an **Argument** of login_cpw.do. The module will present the user with a password change form and have them validate the new password by typing it twice.
Define locations

Locations are used by various applications to locate users, facilities, or configuration items (CI).

You can configure different levels of location in a parent-child hierarchy. For example, an email server might be associated with a location of Second Floor, whereas the email business service might be associated with New York City.

Each level of this hierarchy contains a separate location record, with the next higher level specified as a parent. In this example each location is selectable as a hierarchy from reference fields:
The location is also used to generate a full identifier in the **Full name** field, which is available by configuring the form as follows.

![Figure 95: Location Full name](image)

Locations are stored in the Location [cmn_location] table.

### Define a location

To create a location, navigate to **User Administration > Locations** and click **New**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the location. This is the display value that the system uses when referencing this location on a form.</td>
</tr>
<tr>
<td>Street</td>
<td>The street address of the location.</td>
</tr>
<tr>
<td>City</td>
<td>The city of the location.</td>
</tr>
<tr>
<td>State / Province</td>
<td>State or province of the location.</td>
</tr>
<tr>
<td>Zip / Postal Code</td>
<td>The zip or postal code of the location.</td>
</tr>
<tr>
<td>Country</td>
<td>The country of the location.</td>
</tr>
<tr>
<td>Contact</td>
<td>Name of a user who is the contact for this location.</td>
</tr>
<tr>
<td>Phone</td>
<td>The phone number for the location.</td>
</tr>
<tr>
<td>Fax phone</td>
<td>The fax number for the location.</td>
</tr>
<tr>
<td>Parent</td>
<td>Name of the parent location for this location. Location hierarchies are described above.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Latitude</td>
<td>The latitude of the location. This field and the Longitude field described below are populated automatically by the get_lat_long business rule when the form contains enough information, such as an address or city name and a postal code. Deactivate this business rule to prevent the system from overwriting any values entered manually into these fields.</td>
</tr>
<tr>
<td>Longitude</td>
<td>The longitude of the location.</td>
</tr>
<tr>
<td>Company</td>
<td>A reference field to the Company [cmn_company] table.</td>
</tr>
<tr>
<td>Full name</td>
<td>A read-only, calculated field that assembles the parent hierarchy of the location into a full name.</td>
</tr>
<tr>
<td>Stock room</td>
<td>A boolean field that identifies whether the location is being used as a stock room.</td>
</tr>
<tr>
<td>Time zone</td>
<td>The location’s time zone. By default, the location uses the system time zone.</td>
</tr>
</tbody>
</table>

**Map location**

The latitude and longitude fields are populated by a business rule (get_lat_long) which queries the Google Map service. The more specific the location is, the more accurate the latitude and longitude are.

After the latitude and longitude are populated, Map Pages can be defined that display locations in an interactive map. For more information, see Map pages on page 246.

- Latitude and longitude are expressed as a floating point data type. Previous releases expressed this information as a string. During an upgrade, the system converts the data where possible.

**Database storage for audio files**

Administrators and users with the image_admin role can upload and store audio files in the database. These audio files can then be referenced in HTML.

An administrator must activate this feature.

**Activate database storage for audio files**

You can activate the Database Storage for Audio Files (com.glide.db_audio) plugin.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
If the plugin depends on other plugins, these plugins are listed along with their activation status.

3. Optional: If available, select the **Load demo data** check box.

Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click **Activate**.

---

### Upload an audio file

You can upload `.mp3` or `.ogg` audio files.

**Role required:** image_admin

1. Navigate to **System UI > Audio Files**.
2. Click **New**.
3. Enter a **Name** for the audio file, including the file extension.
4. Add the **Audio** file.
5. Save the record.

---

### Audio file references in HTML

You can use HTML to reference audio files you have uploaded.

Use the `<audio>` HTML tag and set the `src` attribute to the name of the file as set by the **Name** field of the Audio record. You must use the `controls` attribute to display the audio controls. For example, use the following HTML.

```html
<audio src="Beep.mp3" title="Beep" control="control"/>
```

---

### Storing images in the database

Administrators and users with the image_admin or content_admin role can upload and store images in the database.

These images are saved in the Images `[db_image]` table. Uploading an image to the database allows it to be referenced from HTML fields and by appending the name of the image to the URL of the instance.

#### Images vs attachments

If you want to access an image from a record, or if you want to prevent users from appending the image name to the URL of the instance, upload it as an attachment instead. When you upload an image as an attachment, the image is saved in the Attachments `[sys_attachment]` table. See **Attachment administration** on page 749 and **Manage attachments** on page 128 for more information.

---

### Uploading Images

Navigate to **System UI > Images** to see the list of images stored in the database. Click **New** to add a new image.
The following fields are available:

- **Name**: enter the file name by which to reference the image in HTML.
- **Active**: select the check box to allow the image to be referenced in HTML.
- **Category**: select a category in which to organize the image file.
- **Image**: select **Click to add...** to upload an image. If an image is already uploaded, click [Update] to upload a new version of the image or [Delete] to remove the image.
- **Format, Size bytes, Height, and Width**: view current image metadata; this information is automatically populated when the image is uploaded.

**Figure 96: Upload image**
Acceptable image file types

Upload image files with the following extensions:

- .gif
- .jpg
- .png
- .bmp

**Note:** The system does not support uploading zip files that contain .bmp images.

If you upload an image that may not be supported in Internet Explorer browsers, a warning message appears. For example, Internet Explorer does not support JPEG images that are encoded in CMYK format.

![Image warning](https://via.placeholder.com/150)

Figure 97: Image warning

The system does not support uploading images in the .ico format through this interface. The .ico format is required by Microsoft Internet Explorer for favorite icons. To upload an .ico image, attach the .ico image to a record instead of using the standard image upload interface.

To understand how to insert an image into a form record, see the *Insert an image into your article, incident, or other form record* blog post in the ServiceNow Community.

Referencing an image in HTML

To reference an image in HTML, append an "x" to the file extension.

For example:

- browser_controls.gifx
- browser_controls.jpgx
Updating an existing image

You can change an existing image to an updated version.

Role required: image_admin or content_admin

1. Navigate to System UI > Images.
2. In the Images list, click the name of the image to be replaced.
3. In the Image field, click the [Update] link.
4. Click Browse and navigate to the desired image file.
5. Click Open, and then click OK.

The new file is uploaded into the instance database, replacing the previous file. It will be used in all locations that reference the file name.

[Note:] If the new image does not appear as expected, clear the browser cache.

Upload multiple images

You can upload multiple images at one time.

Role required: image_admin or content_admin

1. Create a .zip file that contains only acceptable image file types.

[Note:] The system does not support uploading zip files that contain .bmp images.

3. Click Choose File then select the file.
4. Click Upload.
A message indicates that the files are uploaded to the database, and the images list is sorted by updated date in descending order (the uploaded images appear first).

**Note:**
- If the .zip file contains a folder structure, the resulting image names are the path name with underscores in place of the slashes. For example, if the .zip contains `myimages/image1.gif` then the resulting image is named `myimages_image1.gif`. 
• Image names cannot exceed 100 characters in length, including folder structure. Files with excessively long names are rejected.
• Uploads are logged in the system. View system logs to see whether files are uploaded or rejected.

User preferences

Individual users can configure many UI features, such as the number of rows per page in a list or whether the response time displays at the bottom of a list or form.

These user customizations are stored as records in the User Preference [sys_user_preference] table, and are updated each time the user changes the setting.

The UI displays according to each user's preferences.

For example, by default the response time may appear at the bottom of lists and forms. If a user hides the response time, a user preference record is created showing the response time indicator as hidden. During the user's future sessions, the response time indicator is hidden. If the same user subsequently displays the response time, the user preference record is updated appropriately and future sessions open with the response time indicator visible.

Navigate to User Administration > User Preferences for a list of user preference records. Click a preference name to display that preference in form view.

![User Preference form](image)

Figure 98: User Preferences form

Each user preference record includes these fields.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the feature or functionality.</td>
</tr>
<tr>
<td>Description</td>
<td>An optional short description of the feature or functionality.</td>
</tr>
<tr>
<td>System</td>
<td>Shows whether this record indicates the system-wide default (TRUE), or not (FALSE).</td>
</tr>
<tr>
<td>Type</td>
<td>Shows the data type of entry accepted for the Value. For example, you can select string or integer.</td>
</tr>
<tr>
<td>User</td>
<td>Shows the name of the user for whom the setting is customized. If User is blank, the record is for a system-wide default.</td>
</tr>
<tr>
<td>Value</td>
<td>The current setting for this record. Compare this to the User field and System field to determine whether the value shown is a system-wide default or a specific user’s preference.</td>
</tr>
</tbody>
</table>

Records for the system-wide value, which applies to users who have not customized the feature, have these values: **System=True** and **User=blank**. For each user who customizes the features, a separate record is created with these values: **System=False** and **User=<username>**. As a result, the same customizable UI feature may have multiple user preference records.

For some features, the system-wide record does not appear in the User Preferences module initially. The system-wide record may be added to the User Preferences list when a user record is created for the same feature.

When an administrator manually changes a user’s preference value through this module, the user’s next session uses the administrator’s setting. However, the user can customize the features again through the UI, which updates their user preference record. If the administrator deletes the user preference record for a particular user, the user’s next session uses the system-wide value for that feature. When the user subsequently customizes the feature, the system creates a new user preference record for the user.

**User preference settings**

User preferences primarily track the way individual users interact with various features so that new sessions activate the user’s last settings. For example, user preferences track whether the user activates the tabbed or scrolling interface for multi-section forms. Other user preference records allow administrators to adjust certain feature settings.

**View settings**

To view the user preference settings, navigate to **User Administration > User Preferences**.

**User preference list**

This list describes user preference records in the base system. It provides the default value and a description for each. The Updated By column indicates how the preference is set.
<table>
<thead>
<tr>
<th>User preference record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Only</td>
<td>The value is set and updated by user action. Manual changes through the User Preference record do not affect the user experience.</td>
</tr>
<tr>
<td>Admin Only</td>
<td>The administrator can change the value through the User Preference record to modify the user experience.</td>
</tr>
<tr>
<td>Admin or User</td>
<td>The value is set and updated by user action, but the administrator can change the user experience by manually updating the User Preference record.</td>
</tr>
<tr>
<td>System Only</td>
<td>The value is set and updated by the system. Administrators should not modify these records.</td>
</tr>
</tbody>
</table>

**Table 55: User preferences**

<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| `<table>.autocomplete.contains` | Search   | Admin Only | If the glide.ui.ref_ac.startswith property (System Properties > All Properties) is set to `false`, autocomplete.contains determines whether reference fields for the named table use a `contains` or a `starts with` search to provide auto-complete text.  
**True** = use a `contains` search to auto-complete reference fields.  
**False** = use a `starts with` search to auto-complete reference fields.  
**Type**: True/False  
**Default value**: False |

© 2017 ServiceNow. All rights reserved
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| `<table>.db.order`  | List Sort | Admin or User | For the identified table, indicates which column is used to sort the records in a list. This record updates automatically each time the user changes the sort order for the table.  
**Type:** String  
**Default value:** User selection or the `Number` column if the user does not select a column. If the `Number` column is empty, the `Name` field is used to sort the records. |
| `<table>.db.order.direction` | List Sort | Admin or User | For the identified table, indicates whether the list shows records in ascending (ASC) or descending (DESC) order by the field identified in `<table>.db.order`. This record updates each time the user changes the sort direction for the table.  
**Type:** String  
**Default value:** None |
| bsm_map.default_ci  | BSM      | User Only     | Contains the sys_id of the last configuration item the user viewed in the business service management (BSM) map. This record updates automatically each time the user views a new configuration item in the BSM map.  
**Type:** String  
**Default value:** None |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| ci_manage_relationships_filter   | CMDB_hint.cmdb_ci         | Admin Only     | Constrains what appears in the **Available CIs** side of the slush bucket when defining a relationship for a configuration item. The value is in the format of an encoded query string.  
**Type:** String  
**Default value:** locationANYTHING^operational_statusANYTHING |
| ci_manage_relationships_filter   | CMDB_hint.sys_user        | Admin Only     | Constrains what appears in the **Available User** side of the slush bucket when defining a relationship for a configuration item. The value is in the format of an encoded query string.  
**Type:** String  
**Default value:** active=true |
| ci_manage_relationships_filter   | CMDB_hint.sys_user_group  | Admin Only     | Constrains what appears in the **Available Groups** side of the slush bucket when defining a relationship for a configuration item. The value is in the format of an encoded query string.  
**Type:** String  
**Default value:** active=true |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>collapse.&lt;related table&gt;,&lt;related field&gt;</td>
<td>Forms</td>
<td>User Only</td>
<td>Indicates that a related list is collapsed. The same related list will be collapsed when the user's next session begins. This record updates automatically whenever the user expands or collapses the same related list. <strong>True</strong> = collapse the identified related list. <strong>False</strong> = expand the identified related list. <strong>Type</strong>: String <strong>Default value</strong>: None</td>
</tr>
<tr>
<td>collapse.section.&lt;sys_id&gt;</td>
<td>Forms</td>
<td>User Only</td>
<td>Indicates the sys_id of a form section that is collapsed. The same form section will be collapsed when the user's next session begins. This record updates automatically whenever the user expands or collapses the same section of the same form. <strong>True</strong> = collapse the identified section. <strong>False</strong> (or blank) = expand the identified section. <strong>Type</strong>: True/False <strong>Default value</strong>: False or blank</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------</td>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.ui11.use        | UI11     | Admin or User      | Indicates whether UI11 is active for this user. This record updates automatically when the user clicks the **Switch to the new UI** or the **Switch to the old UI** link near the top right of the screen. Manual changes made by the administrator through **User Administration > User Preferences** are implemented in the user's next session.  
**True** = UI11 is active.  
**False** = UI11 is not active.  
**Type**: True/False  
**Default value**: False |
| glide.ui.navpage_state| Menus    | System Only        | **DO NOT MODIFY.** Indicates the user's UI11 configuration.  
**Type**: String  
**Default value**: None |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.javascript_editor</td>
<td>UI</td>
<td>Admin or User</td>
<td>Indicates whether the JavaScript editor and formatting controls are enabled or disabled for the user. The JavaScript editor icon and formatting controls are available only in instances where the Syntax Editor plugin is active. This record updates automatically when a user clicks the icon to enable or disable script editing. For more information, see Syntax editor on page 3816. True = enable script editor. False = disable script editor. Type: True False Default value: True (if syntax editor feature is active)</td>
</tr>
<tr>
<td>glide.ui.response_time</td>
<td>UI</td>
<td>Admin or User</td>
<td>Determines whether the response time information is expanded or collapsed at the bottom of a list or form. This record updates automatically whenever the user clicks the response time indicator icon (clock) at the bottom of a list or form. True = expand the response time information. False = collapse the response time information. Type: String Default value: False</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| homepage                    | Homepages    | Admin or User | Indicates which homepage appears when a user first logs in to the instance or clicks the homepage icon. The value indicates the sys_id of the selected homepage. For more information, see *Homepage administration* on page 476.  
**Type**: String  
**Default value**: None |
| knowledge.search.sort.field | Last Sort    | Admin or User | Determines the sort order for results when searching the knowledge base. This record updates each time the user selects a different Sort by option in the search results header. Available options are: Number of Views (views), Relevancy (relevancy), and Last Modified (sys_updated_on).  
**Type**: String  
**Default value**: views |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_edit_double</td>
<td>Lists</td>
<td>Admin or User</td>
<td>Indicates what action opens the list editor. This record updates automatically when the user personalizes any list (gear icon above the first column in the list) and changes the double click to edit setting. For more information, see List editor administration on page 681. True = open the list editor when the user double-clicks a field in a list. False = open the list editor when the user single-clicks a field in a list. Type: True/False Default value: True</td>
</tr>
<tr>
<td>list_edit_enable</td>
<td>Lists</td>
<td>Admin or User</td>
<td>Indicates whether the list editor is available for editing fields directly in a list. This record updates automatically when the user personalizes any list (gear icon above the first column in the list) and changes the Enable list edit setting. True = enable use of the list editor. False = disable use of the list editor. Type: True/False Default value: True</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>live_message.feed.last_display</td>
<td>Preference</td>
<td>Admin or User</td>
<td>Stores the last feed the user viewed: <strong>My Feed</strong> <em>(sysparm_feed_type=myFeed)</em> or <strong>Company Feed</strong> <em>(sysparm_feed_type=company_feed)</em>. This record updates automatically each time the user selects a different feed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Type:</strong> String</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Default value:</strong> sysparm_feed_type=company_feed</td>
</tr>
<tr>
<td>live_tag.feed.order</td>
<td>Live</td>
<td>Admin or User</td>
<td>Tracks the user's choice of sorting for tag display. This record updates automatically when a user selects a different <strong>Tag Option</strong> in Live. Available options are <strong>Order by Created</strong> <em>(sys_created_on)</em> and <strong>Order by Last Activity</strong> <em>(last_activity)</em>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Type:</strong> String</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Default value:</strong> None (defaults to sys_created_on)</td>
</tr>
<tr>
<td>menu.&lt;identifier&gt;.expanded</td>
<td>Menu</td>
<td>User Only</td>
<td>Indicates the sys_id of a section of the application navigator menu that is open (expanded) for the user. The same section will be open when the user's next session begins. This record updates automatically each time the user expands or collapses the same section of the application navigator.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Type:</strong> String</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Default value:</strong> None</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| mobile_rowcount  | Mobile   | Admin or User    | Indicates the number of rows displayed in a list on a mobile device. This record updates automatically each time the user chooses a different number of rows per page from the mobile screen. For more information, see View or modify a mobile list on page 1167.  
**Type:** Integer  
**Default value:** 20 |
| mobile_use_full.android | Mobile   | Admin or User    | Indicates which user interface appears on the user’s Android mobile device. This record updates automatically each time the user clicks the gear icon above the mobile homepage to display the full desktop interface or clicks the cell phone icon in the welcome bar to display the mobile interface. For more information, see Smartphone interface on page 1158.  
**True** = display the full desktop interface on Android mobile devices.  
**False** = display the mobile interface on Android mobile devices.  
**Type:** True/False  
**Default value:** False |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| mobile_use_full.iphone      | Mobile   | Admin or User| Indicates which user interface appears on the user's iPhone mobile device. This record updates automatically each time the user clicks the gear button above the mobile homepage to display the full desktop interface or clicks the cell phone icon in the welcome bar to display the mobile interface.  
**True** = display the full desktop interface on an iPhone mobile device.  
**False** = display the mobile user interface on an iPhone mobile device.  
**Type**: True/False  
**Default value**: False |
| module                      | Menus    | User Only    | Records the sys_id of the last module the user accessed.                                                                                                                                                 |
|                             |          |              | **Type**: String  
**Default value**: sys_id                                                                                                                                  |
| owned_by_indicator.form     | Update Sets | Admin Only  | Shows or hides the update indicator icon in form headers when customer updates to that form are being tracked by update sets.  
**True** = show the customer updates indicator icon.  
**False** = hide the customer updates indicator icon.  
**Type**: True/False  
**Default value**: False |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>recent.impersonations</td>
<td>Administration</td>
<td>User Only</td>
<td><strong>DO NOT MODIFY.</strong> Shows who the administrator most recently impersonated. For more information, see <em>Impersonate a user</em> on page 1759. <strong>Type:</strong> String <strong>Default value:</strong> None</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>report.expanded</td>
<td>Reporting</td>
<td>Admin or User</td>
<td>Works in combination with the UI property List of roles (comma-separated) that can expand the report header to determine whether the report header is open when viewing a report. For users whose role permits them to access the report header, the report.expanded user preference indicates whether reports open with the header expanded or collapsed. Expanding the report header reveals the report builder which is used to modify the report definition. This user preference record updates automatically each time the user expands or collapses the report header. If the user's role does not have permission to view the report header, the report header is never accessible, regardless of this user preference setting. True = expand the report header when displaying a report. False = collapse the report header when displaying a report. Type: String Default value: True</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>rowcount</td>
<td>Lists</td>
<td>User Only</td>
<td>Indicates the maximum number of rows that display on a single page in a list. This value also determines the maximum number of records that display in a list gauge on a homepage. This record updates automatically when a user chooses a different number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Type:</strong> Integer</td>
</tr>
<tr>
<td>sys_update_set</td>
<td>Update Sets</td>
<td>User Only</td>
<td>DO NOT MODIFY. Indicates the update set that is currently active. This value update automatically when a user selects a different update set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Type:</strong> GUID</td>
</tr>
<tr>
<td>tabbed.forms</td>
<td>Forms</td>
<td>User Only</td>
<td>Indicates whether forms that contain more than two sections use a tabbed interface or a scrolling list of sections. This setting applies to all forms; it is not set on a form-by-form basis. This record updates automatically when a user clicks the <strong>Toggle Tabs</strong> icon in the banner bar. For more information, see <strong>Tabbed forms</strong> on page 744.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>True</strong> = display multi-section forms as tabs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Type:</strong> True/False</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>table.compact</td>
<td>Lists</td>
<td>User Only</td>
<td>Indicates whether lists appear with more or less blank space above and below each row. This setting applies to all lists; it is not set on a list-by-list basis. This record updates automatically when a user personalizes any list and changes the Compact rows option. True = reduce the blank space above and below each row in a list to show more rows on the screen at one time. False = add blank space above and below each row in a list to improve readability. Type: True/False Default value: False</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved. 226
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| table.wrap     | Lists    | User Only    | Indicates whether long text in a list is wrapped onto multiple lines or truncated. This setting applies to all lists; it is not set on a list-by-list basis. This record updates automatically when a user personalizes any list (gear icon above the first column in a list) and changes the Wrap column text option.  
**True** = wrap long text in a list. All text is seen in list view, but each row may occupy more vertical space.  
**False** = do not wrap the long text in a list. Text is truncated in list view, but each row takes less vertical space. Full text can be seen in form view.  
**Type**: True/False  
**Default value**: True |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| top_searches.period   | Homepages     | User Only  | Indicates the default time period (hour, day, week, or month) covered by top searches widgets on a homepage. When this record exists for individual users, it indicates the current time period for the Top Searches - All widget. The time period setting for top searches widgets on individual tables is stored in the top_searches.period.<table> user preference. For more information, see Top Searches homepage on page 480.  
**Type**: String  
**Default value**: Day |
| top_searches.period.<table> | Homepages     | User Only  | Indicates the time period (hour, day, week, or month) covered by top searches widgets for the specified table. This record updates automatically when a user chooses a different time period in a top searches widget for the specified table. If no record exists for a specific table, top searches widgets for that table use the time period set in the top_searches.period user preference where **System=true**.  
**Type**: String  
**Default value**: Day |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts.match</td>
<td>Text Search</td>
<td>Admin or User</td>
<td>Indicates whether the task record should be returned by a global text search in which the search text exactly matches a task number. This record updates automatically when a user clicks the Search tips and preferences link on the search results page and changes the Return task record if searching for exact number setting. For more information, see Global search user preferences on page 1078. True = return the task record. A link is provided for full search results. False = return full search results, even if the search term matches a task number. Type: True/False Default value: True</td>
</tr>
<tr>
<td>Preference</td>
<td>Category</td>
<td>Updated by</td>
<td>Details</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ts.remember.expanded| Text Search   | Admin or User    | Indicates whether to remember which search groups were expanded and collapsed during the previous search. The search groups available depend on the user's access rights and selections. This record updates automatically when the user clicks the **Search tips and preferences** link on the search results page and changes the **Use remembered expand / collapse preferences** setting.  
**True** = remember the expand/collapse setting and use it for subsequent searches.  
**False** = do not remember the expand/collapse setting. Expand all groups for subsequent searches.  
**Type**: String  
**Default value**: True |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| ts.show_empty_groups| Text Search    | Admin or User  | Indicates whether global text search results include groups with no matches. The search groups available depend on the user's access rights. This record updates automatically when the user clicks the Search tips and preferences link on the search results page and changes the Show groups with no search matches setting.  
**True** = include empty groups when displaying global text search results.  
**False** = hide empty groups when displaying global text search results.  
**Type**: String  
**Default value**: True |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| ts.show_negative_result_info   | Text Search   | Admin or User      | Indicates whether group headers in the global text search results page should identify tables that had no matches. The search groups available depend on the user's access rights. This record updates automatically when the user clicks the **Search tips and preferences** link on the search results page and changes the **In search group header, list tables with no search matches** setting.  

**True** = show tables that returned no matches in the group header.  

**False** = hide tables that returned no matches in the group header.  

**Type**: String  

**Default value**: True  


<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| ts.show_search_groups | Text Search     | Admin or User  | Indicates whether the global text search results page includes check boxes that let the user indicate which groups to search and display. The search groups available depend on the user’s access rights and settings. This record updates automatically when the user clicks the **Search tips and preferences** link on the search results page and changes the **Show selectable search groups** setting.  
  
  **True** = show the check boxes for enabling or disabling individual search groups.  
  
  **False** = hide the check boxes for enabling or disabling individual search groups.  
  
  **Type**: String  
  
  **Default value**: True |
| user.can.logout       | Security        | Admin Only     | Indicates whether users see a **Logout** button. If this is False, users are automatically logged out when their session times out. For more information, see Remove the Logout button on page 2479. This user preference does not apply to UI16.  
  
  **True** = show the **Logout** button, which enables manual logout.  
  
  **False** = hide the **Logout** button, which disables manual logout.  
  
  **Type**: True/False  
  
  **Default value**: True |
<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated by</th>
<th>Details</th>
</tr>
</thead>
</table>
| use.concourse    | UI16     | Admin Only        | Indicates whether users see UI15 or UI16. The UI16 plugin [com.glide.ui.ui16] must be enabled for users to see UI16.  
**True** = show UI16 by default  
**False** = show UI15 by default  
**Type:** String  
**Default value:** True |

User preferences and update sets

User preference records for system-wide values, also called the default or global values, are stored in update sets.

Any changes are implemented when you import the update set and affect all users who have not customized the feature. User preference records for specific users are not stored in update sets, so user customizations are retained when you import an update set.

Troubleshooting user-specific UI differences

If an individual user encounters an unexplained behavior in the user interface, you can check their user preferences.

Role required: admin

1. Navigate to **User Administration > User Preferences**.
2. Search for the user name to find all that user's records.
3. Delete the record that affects the behavior in question.

   The system-wide preference is active during the user's next session. If the user customizes the behavior, a new user preference record is created and used for subsequent sessions.

View management

A view defines the elements that appear when a user opens a form or a list.

Administrators and users with the personalize role can define views for any list and form, which view should be visible by default, and which views pertain to specific user roles. Users with the admin or view_changer roles can change views.

For example, this is the Incident form in the **Self-Service View**:
Figure 99: Self service view

This is the Incident form in the **Metrics View**:
Figure 100: Metrics view

**Note:** Views can be used to define base views. UI Policies can modify those views based on context. For more information, see *Create a UI policy* on page 753.

**Administer views**

Several views are included with the base system, including the Default view and Advanced view.

**Warning:** Do not delete any of the base system views.

All view records are saved in the UI View [sys_ui_view] table. Every list and form either has a view associated with it or uses the Default view if no other view is assigned. Administrators can create additional views or modify the base system views.

**Switch views**

You can switch the view from the default for lists and forms.
To switch between list views, click the table name at the top left corner of the list, and then select Views > [Desired View]:

![List view dropdown](image)

Figure 101: List view dropdown

To switch between form views, click the table name at the left side of the form header, and then select Views > [Desired View]:

![Form view dropdown](image)

Figure 102: Form view dropdown

Switching views submits the form, which saves all changes and triggers any onSubmit client scripts that apply. You cannot switch form views on a new form that has not been saved yet.

When a user switches views, the selected view is saved as a user preference so the user sees the same view by default when the form opens. When a user has a view saved as a user preference and then opens a URL to a record that specifies another view, the form displays in the view saved in the user preference, not the URL. For example, if a user selects the Mobile view on an Incident record and then tries to open the following link, which specifies the visual task board view, the form still opens in the Mobile view:

https://{instance}/nav_to.do?uri=incident.do?sys_id={sys_ID}sysparm_view=vtb

The sysparm_view parameter specifies the view to be used for a list or a form, and can be overwritten by a user’s stored preference for a view. You can override this behavior by setting the sysparm_view_forced parameter to true.

**Create a view**

You can create views to force all users to use a certain view, despite the view specified by the user’s preference.

Role required: admin

1. Navigate to the list or form that you want to create the view for.
2. Right click the header and select Configure > List Layout or Configure > Form Layout.
3. Under the List View section, select the view on which you want to base your new view.
   The fields visible for that view appear in the Selected list.
4. From the choice list, select New.
   The Create New View form appears.
5. Enter the descriptive name of the view. View names cannot use special characters or spaces, only the characters A-Z, a-z, 0-9 and _.

6. Click OK. The fields in the Available column are the same as the first view you based the new view on.

7. Select the fields to appear in this view by adding or removing the fields from the Selected column. You can also adjust the order they appear on the form by moving the fields up or down. If you are creating a view for a form, you can select a form section and configure the fields for that section. You can also create views in the same manner when you configure a related list.

Delete a view
You can delete any view that you created.
Role required: admin
Do not delete the base system views.

1. Navigate to System UI > Views.
2. Click the view to delete.
3. Click Delete on the form header.

Create a view rule
When a user switches views, the selected view is saved as a user preference so the user sees the same view by default when the form opens. You can override this functionality to force a specified view to be used.
Role required: admin
View rules do not apply to users who have no role.

1. Navigate to System UI > View Rules.
2. Click New.
3. Complete the form.

### Table 56: View rule form

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify an identifying name for the rule. This field is case-sensitive and must match exactly the view name.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this option to apply the view rule according to the conditions you specified. If unchecked, the view rule is not be applied.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this option to specify a code-based condition instead of using the condition builder. Selecting this option displays the Script field and hides the Match conditions, Conditions, and View fields.</td>
</tr>
<tr>
<td>Match Conditions</td>
<td>Select whether Any or All of the conditions need to be met. The system hides this field when you create an advanced View Rule.</td>
</tr>
</tbody>
</table>
### View rule form

The View Rule [sysrule_view] table contains the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditions</td>
<td>Use the condition builder to determine when the view is applied. The system hides this field when you create an advanced View Rule.</td>
</tr>
<tr>
<td>Application</td>
<td>Displays the application to which the View Rule record belongs.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the table on which this view rule will be applied. The list shows only tables and database views that are in the same scope as the view rule</td>
</tr>
<tr>
<td>Device type</td>
<td>Select which interface this view rule applies to.</td>
</tr>
<tr>
<td>View</td>
<td>Select the view that is automatically applied if the conditions match. The system hides this field when you create an advanced View Rule.</td>
</tr>
<tr>
<td>Script</td>
<td>Enter a script to determine when to display a particular view. The system only displays this field when you create an advanced View Rule.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

---

© 2017 ServiceNow. All rights reserved.
### Field | Input value
--- | ---
Conditions | Use the condition builder to determine when the view is applied. The system hides this field when you create an advanced View Rule.
Application | Displays the application to which the View Rule record belongs.
Table | Select the table on which this view rule will be applied. The list shows only tables and database views that are in the same scope as the view rule.
Device type | Select which interface this view rule applies to.
View | Select the view that is automatically applied if the conditions match. The system hides this field when you create an advanced View Rule.
Script | Enter a script to determine when to display a particular view. The system only displays this field when you create an advanced View Rule.

---

Restrict view by role

**Caution:** The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

**Name:** Restrict View by Role

**Type:** System UI > View Rules

**Table:** Any

**Description:** Generally the view used on a form is controlled by either specifying the sysparm_view parameter in the url or module properties, or by inheriting the view from the previous form or list. In some cases this does not apply, such as opening a referenced field form from a record producer. In this case you may want to control the view of the form based on roles. This script assumes there is a view called ess available to the current table.

**Parameters:**
- view - A string containing the name of the current view.
- is_list - A Boolean value indicating whether this is a list view.

**Script:**

```javascript
(function overrideView(view, is_list) {
    // Force non-itil users to use the ess view
    if (gs.hasRole("itil")) {
        return;
    }
    if (view.startsWith("ess")) {
        return;
    }
    // do not change view if it starts with sys_ (e.g. sys_ref_list)
    if (view.startsWith("sys_")) {
        return;
    }
});
```
Navigation Handler

A Navigation Handler is essentially a scripted View Rule and runs each time data from the specified table is requested in the form view.

The following script, which is a navigation handler included with the HR plugin, forces the ESS view for users with no roles, and uses the default view for all other users.

```javascript
var gr = new GlideRecord(hr.TABLE_CASE);
if (gr.get(g_uri.get('sys_id'))) {
    if (!gs.getUser().hasRoles())
        g_uri.set('sysparm_view', 'ess');
    else
        g_uri.set('sysparm_view', '');
}
answer = g_uri.toString('hr_case.do');
```

Use the glide.ui.view_rule.check_after_nav_handler system property to control the order in which View Rules and Navigation Handlers are applied. Set the property value to True to process view rules after navigation handlers. If the system property does not exist in your instance, the navigation handler always takes precedence.

The system property only overrides the Navigation Handler if the Navigation Handler scripted function does not return an answer. In the example script above, the property will have no effect as the Navigation Handler will always return an answer due to the answer line being outside of the if statement.

To force the Navigation Handler Script above to honor View Rules for the table, add the property above, set it to true, and update the code to only return an answer when the view needs to be changed or forced.

```javascript
var gr = new GlideRecord(hr.TABLE_CASE);
if (gr.get(g_url.get('sys_id'))) {
    if (!gs.getUser().hasRoles()) {
        g_url.set('sysparm_view', 'ess');
        answer = g_url.toString('hr_case.do');
    }
}
```

Welcome page content

The ServiceNow ITSA Suite login page content is completely customizable by the administrator of the system.

The login page should contain instructions and any other important information you would like to convey to your customers each time they use the system.

Login language selection

Users can select their language on the login page.

This enables them to start their user session in the language of their choice.
Create company-specific welcome page content

You can create company-specific welcome page content.

Role required: admin

1. Navigate to System UI > Welcome Page Content.
2. To create a new section, click New.
3. Enter the following in the Condition field, where Company is the name of the company whose users see this section.
   
   \[ \text{gs.getUser().getCompanyRecord().name.toString() == "Company"} \]

4. Complete the form.
5. Click Submit.
Sections

To view and configure the welcome page elements, navigate to System UI > Welcome Page Content. You may need to enable the module if it is not visible. If any of the sections are not needed, they may be removed. You can edit any item by clicking its short description.

- **Demonstration**: Information about logging in to a ServiceNow ITSA Suite demo system
- **Welcome**: General information
- **Using**: Some tips and tricks for using the system
- **More Information**: A link to the ServiceNow wiki
- **Login**: Login information

Google Maps integration

Map pages enable you to graphically display data on a Google map page based on location data. Maps can be generated using basic JavaScript, but are flexible enough to display even the most complicated of queries. The maps you generate use standard Google Maps API mapping features, including a variety of link types to records in your instance. This feature requires the Google Maps plugin.

![Google Maps integration](image)

Figure 104: Map page

Google Maps setup

Before using the Google Maps API, users may need to enter a Google license key in system properties, and configure other options.

You may need to obtain a Google Maps for Work license key to cover development use of the Google Maps API. Please see the [Getting Started Guide](#) on the Google web site for details on obtaining a key, using the API, and relevant terms of service.

When you receive your key, enter it in the System Properties > Google Maps property form, and configure your maps using the configuration options in this screen.
After you complete the configuration, you can create new map pages or use the default pages included with the plugin. Map pages define what data is displayed on the map and the appearance of the links. When you have the map pages you need, you can create new modules in your application navigator to display your maps.

For a tutorial on how to display all markers that link to your records, see the Display Map Markers on Google Maps blog post on the ServiceNow Community.

Google Maps plugin

The Google Maps Plugin provides the following configurable properties in System Properties > Google Maps.

Table 58: Google Maps system properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.maps.auto_close</td>
<td>If true, automatically closes a map information window before opening a new one.</td>
</tr>
<tr>
<td></td>
<td>• Type: true/false</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.client</td>
<td>Client ID for Google Maps API for Work.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: gme-servicenow</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.private.key</td>
<td>Private key for Google Maps API for Work. This key activates the geolocation feature, which locates users in the system precisely, using data from their mobile devices.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: empty</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.version</td>
<td>Version number of the current installation of Google Maps API.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: current version number</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>google.maps.key</td>
<td>The Google Maps API key that is tied to the URL of the server. This key authorizes development use of Google Maps API.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: empty</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.latitude</td>
<td>Starting latitude of the map. This value determines the starting position displayed in Google Maps.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: 36.008522</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.longitude</td>
<td>Starting longitude of the map. This value determines the starting position displayed in Google Maps pages.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: -95.221764</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.max_items</td>
<td>Maximum number of items to display on the map.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 500</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.table</td>
<td>Table used by the map. The table needs the following fields: name, longitude, latitude.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: cmn_location</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.zoom</td>
<td>Starting zoom level of the map (1 is the lowest)</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: 4</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
</tbody>
</table>
Map pages

Map pages display ServiceNow data graphically on a Google map page based on location data that you provide.

You can create as many map pages as needed to define the types of data to display, the links to show, and the appearance of the map. After creating the map pages, you can create new modules to display the map pages.

**Note:** Map pages are subject to the non-production access limitations. See *Google Maps integration* on page 243.

Create a map page

Your ServiceNow instance provides several default map pages. You can also create custom map pages with the script examples described in this page.

Role required: admin

The default map page record displays critical incidents on a Google map page.
1. Navigate to **System UI > Map Pages**.
2. Click **New**.
3. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name to identify the map.</td>
</tr>
<tr>
<td>Center map on</td>
<td>Enter the location for centering the map using the decimal form of latitude and longitude. For example, to center the map on the statue of liberty, enter 40.689865, -74.045235.</td>
</tr>
<tr>
<td>Initial zoom</td>
<td>Set the map’s zoom level on opening.</td>
</tr>
<tr>
<td>Controls size</td>
<td>Select whether the Google navigation icons on the map should be large or small.</td>
</tr>
<tr>
<td>Type</td>
<td>Select a map background style: <strong>Satellite</strong>, <strong>Hybrid</strong>, <strong>Normal</strong>, or <strong>Terrain</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type selection</td>
<td>Select which type of Google icons to display on the map: <strong>Buttons</strong> or <strong>Menu</strong>.</td>
</tr>
<tr>
<td>Map overview</td>
<td>Enable or disable a wide view map window for navigation.</td>
</tr>
<tr>
<td>Script</td>
<td>Create a script to define the type of data to display on this map. For more information, see <em>Scripting for map pages</em> on page 252.</td>
</tr>
</tbody>
</table>

Fields you can add by configuring the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center latitude</td>
<td>The latitude the map page will center on. This field is populated by the business rule Geocode Address.</td>
</tr>
<tr>
<td>Center longitude</td>
<td>The longitude the map page will center on. This field is populated by the business rule Geocode Address.</td>
</tr>
</tbody>
</table>

4. Save the record.

**Map page modules**

You can create a new application module for desktop or smartphone users to display a Google map page.

You can create a desktop application module for a Google map page using either of the following methods.

- Map page link
- URL link

**Create a desktop map page module using a map page link**

You can create a custom desktop application module that links directly to a map page.

Role required: admin

1. Select the appropriate application for the module.
   
   For example, if you are planning to display critical incidents, add a module to the Incident application menu.

2. Perform the appropriate action for your version of the UI:

   **UI16**
   
   1. Navigate to **System Definition > Application Menus**.
   2. Open the application menu to which you want to add the module.

   **UI15 or UI11**
   
   Right-click the application menu in the application navigator and select **Edit Application Menu**.

3. In the **Modules** related list, click **New**.

4. Complete the form, as appropriate.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Enter a name for the module in the application navigator.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to define the sequence this condition should be evaluated if more than one matching condition exists. The order is evaluated from the lowest value to the highest value.</td>
</tr>
<tr>
<td>Application menu</td>
<td>Select the application menu where you want this module to appear. By default, this field displays the application menu that you opened to create the module.</td>
</tr>
<tr>
<td>Hint</td>
<td>Enter a brief description to display when a user points to the module name in the application navigator.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate this module. Only active modules appear in the application menu.</td>
</tr>
<tr>
<td>Image</td>
<td>Select an icon to display with this module in the application navigator. If this field is blank, a default icon is used.</td>
</tr>
<tr>
<td>Link type</td>
<td>Select Map Page from the list. This selection changes the available fields in the form.</td>
</tr>
<tr>
<td>Map page</td>
<td>Select the pre-configured map page to use for this module. For example, Critical Incidents.</td>
</tr>
<tr>
<td>Roles</td>
<td>Select the roles that are permitted to access this module. If this field is blank, all roles can access the module.</td>
</tr>
</tbody>
</table>
Create a desktop map page module using a URL link

You can create a desktop module that links to a map page via a URL.

Role required: admin

1. Select the appropriate application for the module.
   
   For example, if you are planning to display critical incidents, add a module to the Incident application menu.

2. Perform the appropriate action for your version of the UI:

<table>
<thead>
<tr>
<th>UI16</th>
<th>UI15 or UI11</th>
</tr>
</thead>
</table>
   | 1. Navigate to **System Definition > Application Menus**.  
     2. Open the application menu to which you want to add the module. |
   | Right-click the application menu in the application navigator and select **Edit Application Menu**. |

3. In the **Modules** related list, click **New**.

4. Select a **Link type** of **URL (from Arguments:)**. This selection displays the Arguments field.

5. Enter the URL in this format: `map_page.do?sysparm_name=<map page name>`.

**Note:** If the map page title has a space in it, replace the space with `%20` for the correct syntax. For example, a map page called Critical Incidents becomes `Critical%20Incidents%20` in a URL.
6. Click Submit.

Create a smartphone map page module

You can create a custom smartphone module that links directly to a map page.

Role required: admin

1. Obtain the sys_id of the map page.
   The sys_id is required when you create the module.
2. Navigate to System Mobile UI > Navigator Apps.
3. Open the application menu where you want the new module to appear.
4. In the Modules related list, click New.
5. Complete the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the module.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to specify the order of the module within the application menu. For example, an entry of 100 would place this module before one with an Order entry of 200.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application menu</td>
<td>Select the application menu where you want this module to appear. By default, this field displays the application menu that you opened to create the module.</td>
</tr>
<tr>
<td>Table</td>
<td>Do not select a table.</td>
</tr>
<tr>
<td>Updated</td>
<td>Displays the date and time when the module record is updated.</td>
</tr>
<tr>
<td>Roles</td>
<td>Select the roles that are permitted to access this module. If this field is blank, all roles can access the module.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate this module. Only active modules appear in the application menu.</td>
</tr>
<tr>
<td>Filter</td>
<td>Do not add a filter condition.</td>
</tr>
<tr>
<td>Path</td>
<td>Enter map/ followed by the sys_id of the map page. For example: map/c86c5feac0a80a6600706f0102968196</td>
</tr>
</tbody>
</table>

6. Click Submit.

Scripting for map pages

In the Script field on the Map Page form, you can use the available attributes or custom code to define map characteristics, such as marker appearance for items on the map, what information to display when someone clicks a marker, and more.

You can use the isMobile variable to set custom behavior for the smartphone view of the map.

To create an item on the map, use the `map.addItem(glideRecord)` method. You must pass a valid GlideRecord to addItem().

The following attributes are available.

Table 59: Scripting map item attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name used for identification.</td>
</tr>
<tr>
<td>latitude</td>
<td>Latitude is not necessary if you define an address.</td>
</tr>
<tr>
<td>longitude</td>
<td>Longitude is not necessary if you define an address.</td>
</tr>
<tr>
<td>icon</td>
<td>URL of the icon to display for the marker. The default Google marker is used if a custom icon is not specified.</td>
</tr>
<tr>
<td>icon_width</td>
<td>Width of the icon. The default is 32.</td>
</tr>
<tr>
<td>icon_height</td>
<td>Height of the icon. The default is 32.</td>
</tr>
</tbody>
</table>
### Attribute Description

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table_name</td>
<td>Table whose records display when the marker icon is clicked. Used in conjunction with the sys_id attribute.</td>
</tr>
<tr>
<td>sys_id</td>
<td>Sys_id of the record that will display when the marker icon is clicked. Used in conjunction with the table_name attribute.</td>
</tr>
<tr>
<td>view</td>
<td>View of the form displayed in the dialog box when the marker icon is clicked.</td>
</tr>
<tr>
<td>dialog_title</td>
<td>Title of the dialog box containing the record information.</td>
</tr>
<tr>
<td>html</td>
<td>Arbitrary HTML code for the pop-up window. If used, this value overrides the dialog box.</td>
</tr>
<tr>
<td>marker_label</td>
<td>Optional marker icon label text. For an example, see Map page marker label script on page 255.</td>
</tr>
<tr>
<td>label_offset_left</td>
<td>Optional attribute that is used with marker_label to define the horizontal position of the marker label. The default is 0.</td>
</tr>
<tr>
<td>label_offset_top</td>
<td>Optional attribute that is used with marker_label to define the vertical position of the marker label. The default is 0.</td>
</tr>
</tbody>
</table>

**Scripting custom map page behavior for smartphone interfaces**

If you plan to access the map page from a smartphone, you may want to set custom smartphone interface behavior using the isMobile variable.

For example, you might set different values for the icon_width and icon_height attributes when isMobile is true.

**Basic map page script**

This script is used to display all active, critical incident locations.

```javascript
//setup new GlideRecord query on the incident table
var gr = new GlideRecord("incident");
//add condition for priority 1
gr.addQuery('priority', '1');
//add condition for active incidents
gr.addActiveQuery();
//execute the query
gr.query();

//loop through the list of incidents returned by the query
while (gr.next()) {

    //create a new map item to display - linked to the current incident record
    var item = map.addItem(gr);
    //add the latitude value from the incident's location
    item.latitude = gr.location.latitude;
    //add the longitude value from the incident's location
    item.longitude = gr.location.longitude;
    //add the incident number to the dialog title
    item.dialog_title = gr.getDisplayValue();
}
```

© 2017 ServiceNow. All rights reserved. 253
isMobile map page script

This script is used to display all active, critical incident locations with custom settings for smartphone users.

```javascript
//setup new GlideRecord query on the incident table
var gr = new GlideRecord("incident");
//add condition for priority 1
gr.addQuery('priority', '1');
//add condition for active incidents
gr.addActiveQuery();
//execute the query
gr.query();

//loop through the list of incidents returned by the query
while (gr.next()) {
    //create a new map item to display - linked to the current incident record
    var item = map.addItem(gr);
    //add the latitude value from the incident's location
    item.latitude = gr.location.latitude;
    //add the longitude value from the incident's location
    item.longitude = gr.location.longitude;
    //add the incident number to the dialog title
    item.dialog_title = gr.getDisplayValue();
    //link to the icon image
    item.icon = "http://maps.google.com/mapfiles/kml/pal3/icon51.png";

    //set the icon size (use smaller icons for smartphone users)
    if (isMobile) {
        item.icon_width = "12";
        item.icon_height = "12";
    } else {
        item.icon_width = "16";
        item.icon_height = "16";
    }
}
```

Advanced map page script

This script is used to display the number of open incidents by location.

This script varies the size of the icon based on the number of open incidents for the location. Using the `html` parameter, it also displays the location name and number of incidents, as well as a link to the list of related incidents.

```javascript
//get the instances url so we can link back to it
var uri = gs.getProperty("glide.servlet.uri");
//create an aggregate query on the incident table
var count = new GlideAggregate('incident');
//set condition for active incidents
count.addQuery('active', 'true');
//set aggregate field to location to get count by location
count.addAggregate('COUNT', 'location');
//execute the query
```
count.query();

//loop through the results
while (count.next()) {

    //get the current record's location
    var loc = count.location;
    //get the count of incidents for this location
    var locCount = count.getAggregate('COUNT', 'location');
    //only display location is there are active incidents
    if (locCount > 0) {
        //create new new map item for this location
        var item = map.addItem(count);
        //set lat/long from the location record
        item.latitude = loc.latitude;
        item.longitude = loc.longitude;
        //build the link to the list of incidents for the location
        var link = 'href=' + uri + 'incident_list.do?sysparm_query=active%3Dtrue^location%3D' + loc;
        //build the html value to be displayed when you click the map icon
        item.html='&lt;a ' + link + '>'+ loc.getDisplayValue() + ' (' + locCount + ')&lt;/a&gt;';
        //link to the icon image
        item.icon = "http://maps.google.com/mapfiles/kml/pal3/icon51.png";
        //set the size of the icon based on the number of active incidents
        if (locCount < 5) {
            item.icon_width = "12";
            item.icon_height = "12";
        } else if (locCount < 15) {
            item.icon_width = "16";
            item.icon_height = "16";
        } else {
            item.icon_width = "32";
            item.icon_height = "32";
        }
    }
}

Map page marker label script

Marker labels allow you to add dynamic text to markers.

This example displays the active incident count at each location.
//get the instances url so we can link back to it
var uri = gs.getProperty("glide.servlet.uri");
//create an aggregate query on the incident table
var count = new GlideAggregate('incident');
//set condition for active incidents
count.addQuery('active', 'true');
//set aggregate field to location to get count by location
count.addAggregate('COUNT', 'location');
//execute the query
count.query();

//loop through the results
while (count.next()) {

//get the current record's location
var loc = count.location;
//get the count of incidents for this location
var locCount = count.getAggregate('COUNT', 'location');
//only display location is there are active incidents
if (locCount > 0) {

//create new new map item for this location
var item = map.addItem(count);
//set lat/long from the location record
item.latitude = loc.latitude;
item.longitude = loc.longitude;

//create a marker label with the count
item.marker_label = locCount;
//define label offset for proper position
item.label_offset_left = -4;
item.label_offset_top = -20;

//option to define table and record for label hyperlink
//setting table and sys_id will override the use of html parameter
//item.table = 'cmn_location';
//item.sys_id = loc;

//build the link to the list of incidents for the location
var link = 'href=' + uri + 'incident_list.do?sysparm_query=active%3Dtrue^location%3D' + loc;
//build the html value to be displayed when you click the map icon
item.html= '<a ' + link + '>' + loc.getDisplayValue() + ' (' + locCount + ')</a>';
Theming an instance

An administrator may want to change the styles throughout the instance to change the look and feel of the usual instance interface.

This can be done in a number of different places, often using global CSS or system properties.

Other sources for information on theming include the following.

- To create a themed front-end for users, see Content Management System on page 2040.
- To find out how to modify the banner image and text, see Modify the banner on page 279.
- For styles of individual fields, see Define field styles on page 851.

Beyond theme controls, it is also possible to customize the UI using customized scripts.

Configure logo, colors, and system defaults for UI16

You can use the Basic Configuration UI16 module to brand your instance with your company logo and colors and set basic system defaults. This is the best place to start if you are setting up your instance for the first time or if you have recently enabled UI16.

To prepare for completing basic configuration, gather the following information.

- Obtain the company banner image to use in the header. The image can be high resolution, but when it displays it is scaled based on the aspect ratio. It scales to a maximum of 20px high.
- Get your company's brand color hex or RGB numbers, typically from your marketing department. Use them to decide how to configure the UI background colors.

Role required: admin

Each color selection option provides a color picker to select a color. The text box beside the color picker lets you enter the value of the color as any of the following CSS formats.

- Name: predefined color names, such as red, green, blue
- RGB decimal: RGB(102, 153, 204)
- RGB hex: #223344

Refer to HTML Color Names (W3CSchools) for information about HTML color names.

1. Navigate to System Properties > Basic Configuration UI16
2. Complete the configuration by changing any of the following settings.

<table>
<thead>
<tr>
<th>Table 60: Basic system configuration properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Label</strong></td>
</tr>
<tr>
<td>Page header caption</td>
</tr>
<tr>
<td>Label</td>
</tr>
<tr>
<td>--------------------------------------------</td>
</tr>
<tr>
<td><strong>Browser tab title</strong></td>
</tr>
<tr>
<td><strong>System timezone for all users unless overridden in the user's record</strong></td>
</tr>
<tr>
<td><strong>Banner image</strong></td>
</tr>
<tr>
<td><strong>Date format</strong></td>
</tr>
<tr>
<td><strong>Time format</strong></td>
</tr>
<tr>
<td><strong>Header background color</strong></td>
</tr>
<tr>
<td><strong>Banner text color</strong></td>
</tr>
<tr>
<td><strong>Navigation background color</strong></td>
</tr>
<tr>
<td><strong>Navigation background color</strong></td>
</tr>
<tr>
<td>Label</td>
</tr>
<tr>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Navigation selected tab background color</td>
</tr>
<tr>
<td>Navigation highlight background color</td>
</tr>
<tr>
<td>Navigation separator color</td>
</tr>
<tr>
<td>Label</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Background for Favorites list, history list, and Connect list background</td>
</tr>
<tr>
<td>Module text color for UI16</td>
</tr>
<tr>
<td>Currently selected Navigation tab icon color for UI16</td>
</tr>
</tbody>
</table>
Configure logo, colors, and system defaults

You can use the Basic Configuration module to brand your instance with your company logo and colors and set basic system defaults. This is the best place to start if you are setting up your instance for the first time.

To prepare for completing basic configuration, gather the following information.

- Obtain the company banner image to use in the header. The image can be high resolution, but when it displays it is scaled based on the aspect ratio. It scales to a maximum of 50px high.
- Decide whether to change the banner and list header background color, and the banner text color. If you plan to do so, obtain your company’s brand color hex or RGB numbers. Typically, these come from your marketing department.

Role required: admin

This procedure applies to configuring UI15 and UI11.

1. Navigate to System Properties > Basic Configuration
2. Complete the configuration by changing any of the following settings.

Table 61: Basic system configuration properties

<table>
<thead>
<tr>
<th>Label</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Border color for UI16</td>
<td>css.$navpage-nav-border</td>
<td>Select or enter the color. Also affects the border of the Filter conversations search box in the Connect Chat sidebar.</td>
</tr>
<tr>
<td>Page header caption</td>
<td>glide.product.description</td>
<td>Change the text that appears next to your logo.</td>
</tr>
<tr>
<td>Label</td>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Banner and list caption background color</td>
<td>css.base.color</td>
<td>Click the color picker and select the color to use. You can also enter the name, RGB decimal, or RGB hex number of the color. Refer to <a href="https://www.w3schools.com/html/html_color_names.asp">HTML Color Names (W3CSchools)</a> for information about HTML color names.</td>
</tr>
<tr>
<td>Browser tab title</td>
<td>glide.product.name</td>
<td>Change the text that appears on the browser tab.</td>
</tr>
<tr>
<td>Banner image</td>
<td>glide.product.image</td>
<td>Click + next to the image and upload your logo.</td>
</tr>
<tr>
<td>System timezone for all users unless overridden in the user's record</td>
<td>glide.sys.default.tz</td>
<td>Select the timezone in the choice list. Click <a href="https://www.w3schools.com/html/html_timezones.asp">Configure available time zones</a> to select the time zones that your users can select from in user preferences.</td>
</tr>
<tr>
<td>Date format</td>
<td>glide.sys.date_format</td>
<td>Select the date and time formats from the choice lists.</td>
</tr>
<tr>
<td>Time format</td>
<td>glide.sys.time_format</td>
<td></td>
</tr>
<tr>
<td>Banner text color</td>
<td>css.banner.description.color</td>
<td>Click the color picker and select the color to use. You can also enter the name, RGB decimal, or RGB hex number of the color.</td>
</tr>
</tbody>
</table>

As you make changes, the page refreshes with a preview of the change. Only you see these changes.

3. Click **Save** at the top or bottom of the page.

After you save the configuration changes, all users who select the **System** theme in their UI personalization options see the new configuration colors.

**Menu categories**

Menu categories let you put CSS styles on application labels to make them stand out from other applications.

You navigate to **System Definition > Menu Categories** to view existing menu categories. This CSS style can include border color, text color, background color, and other options.

The base system menu categories include the following.

**Administration**

Applications in this category have a light blue background.
Custom Applications

Applications in this category have a light grey background.
border: 1px solid #96bcdc; background-color: #FBFBFB;

Label

Applications in this category have a blue background.
border-color: blue; background-color: rgb(102, 153, 204); color: white

Maint

Applications in this category have a blue background.
Style: border-color: blue; background-color: rgb(102, 153, 204); color: white

SocialIT

Applications in this category have a blue background.
border-color: blue; background-color: rgb(102, 153, 204); color: white

To add an application to a category, add it in the related list of applications at the bottom of the category record, or else you can specify the category on the application record.

Business service map properties

Administrators can edit the Business Service Map look and feel in the module BSM Map > Properties.
The following properties can be changed.

• How many child nodes to display when too many children exist for parent CI.
• Color used for the collapsed node when too many children exist for parent CI.
• Text style used for collapsed node when too many children exist for parent CI.
• The color of any of the following items.
  • Affect neighbor node
  • Node with an open change request
  • Node with an outage
  • Node with an open incident
  • Recently expanded nodes

Change survey question header colors

All of the lists and forms draw the color of their headers from the base color, which allows theming from the CSS properties.

One notable exception is the survey, which has its own CSS. It is possible to change the CSS by adding a property.

1. Enter sys_properties.form into the navigation filter and press the Enter key.
   A new Property form displays.
2. Name the new property css.list.row.survey.background.color.
3. Enter the hexadecimal value of the desired color into the **Value** field.
4. Click **Submit**. 
   The survey should now use the desired color.

**CSS properties**

Some theming properties may be customized by navigating to **System Properties > CSS**.

Colors are specified using predefined **color names**, RGB decimals, or RGB hexidecimals. The following properties are available through CSS Properties.

- Banner text color
- Banner and list caption background color
- Font used in forms and lists (this is a global font setting)
- Base font size (UI11)
- Button styles (background color, border color, border width, text color)
- Field status indicator colors (including those for Changed, Mandatory Populated, Mandatory Unpopulated, and Read-only)
- List cell vertical alignment
- Navigator menu styles (text font size, background color, text color)
- Header font name and size
- List and form caption color override
- Global text search background color (both for catalog results and knowledge base results)

**CSS theme support**

Themes give the user interface a specific look and feel by using different color combinations.

System administrators can create new themes or customize the existing themes for an instance. To see existing themes, navigate to **System UI > Themes**.

**Activating CSS theme support**

There are different plugins that activate CSS themes support.

Activate one of the following plugins to use themes, based on the version of the UI you use.

- **UI16**: CSS Theme support - UI15 is activated automatically with UI16
- **UI15**: CSS Theme support - UI14 is activated automatically with UI15
- **UI11**: CSS Theme support

See **Activate a plugin** on page 1233.

**Enable the theme picker**

The theme picker adds a field to the system menu (UI16 or UI15) or the banner frame (UI11) that allows the user to quickly switch between CSS themes during the current browser session.

Role required: admin

You can enable the theme picker by activating the ui_theme_changer UI macro.

1. Navigate to **System UI > UI Macros**.
2. Locate the ui_theme_changer UI macro.
3. Set the **Active** field to **true**.
Select a theme

You can switch between different themes for the user interface. Themes are user specific. Each user can select a different theme and the selection is stored as a user preference.

Select a theme by performing the appropriate action for your version of the UI.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI16</td>
<td>Click the gear icon (ー) in the banner frame to access the System Settings window. Select the Theme tab, and then select the theme.</td>
</tr>
<tr>
<td>UI15</td>
<td>Click the gear icon (ー) in the banner frame to access the system menu. Then select a theme from the Theme picker.</td>
</tr>
<tr>
<td>UI11</td>
<td>Select a theme from the Theme picker in the banner frame.</td>
</tr>
</tbody>
</table>

Configure a company theme

You can configure a default CSS theme for a particular company in UI15. The theme appears for all users associated with the company unless they select a different theme.

Role required: personalize_form

Specify a company theme on the company record.

Note: In UI16, the basic configuration is used instead of the theme selected in the company record.

1. Navigate to System Properties > My Company.
2. Configure the form and add the Theme field.
3. In the Theme field, select a theme.
4. Click Update. Users in this company will see the selected theme the next time they log in, unless they have selected a different theme.

Create or customize a theme

You can create new themes or customize existing themes.

Role required: admin

1. Navigate to System UI > Themes.
2. To create a new theme, click New. To customize an existing theme, click the theme name.
3. Enter a name for the theme in the Name field.
4. Add any of the default CSS styles in the CSS field.

In some cases the default style is not the only style definition for an object. In such cases, you can use the CSS important rule format to override any other style definitions with the default definition. For example, this definition overrides the mandatory text color:

```
label.mandatory.text.color: #FF0000 !important.
```
5. Select the **Active** check box.
6. Select one of the following from the **Device** choice list to identify the user interface version for the theme.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser</td>
<td>Identifies a UI11 theme.</td>
</tr>
<tr>
<td>Doctype</td>
<td>Identifies a UI15 theme.</td>
</tr>
<tr>
<td>Concourse</td>
<td>Identifies a UI16 theme.</td>
</tr>
</tbody>
</table>

7. Click **Submit**.

You must refresh the page to see the new theme in the theme picker. The theme picker displays only themes that are compatible with the current version of the UI.

**Default CSS styles**

You can use CSS styles to control the appearance of elements in a theme.

Your version of the UI determines which CSS styles you can use.

**Default CSS styles for UI16**

```css
$search-text-color: #e7e9eb
$navpage-header-bg: #303a46
$navpage-header-color: #ffffff
$navpage-header-divider-color: #455464
$nav-highlight-main: #3D4853
$subnav-background-color: #455464
$navpage-nav-bg: #303a46
$navpage-nav-bg-sub: #455464
$navpage-nav-color-sub: #bec1c6
$navpage-nav-mod-text-hover: #ffffff
$nav-hr-color: #303a46
$nav-highlight-bar-active: #278efc
$nav-highlight-bar-inactive: #828890
$navpage-nav-selected-bg: #4B545F
$navpage-nav-selected-color: #ffffff
$navpage-nav-unselected-color: #bec1c6
$connect-latest-message: #cfd4d8
$nav-timeago-header-color: #303a46
$navpage-nav-app-text: #cfd4d8
$navpage-nav-app-text-hover: #ffffff
$navpage-nav-border: #ddd
$navpage-button-color: #fff
$navpage-button-color-hover: #7EC24F
```

**Default CSS styles for UI15**

```css
base.color: rgb(102, 153, 204)
base.font-family: arial
/color-darkest: #343d47
```
Reset CSS colors

```
$color-darker: #485563
$color-dark: #81878e
$color-light: #bdc0c4
$color-lighter: #e6e8ea
$color-lightest: #ffffff

$color-positive: ${color-green}
$color-warning: ${color-orange}
$color-negative: ${color-red}
$color-info: ${color-blue}

$color-text: ${color-darkest}

$color-accent: ${color-blue}
$color-accent-light: ${color-blue-lighter}

$color-primary: ${color-accent}

# Banner branding styles
banner.background.attachment: scroll
banner.background.color: #FFF
banner.background.image:url("../images/s.gifx")
banner.background.position: center bottom
banner.background.repeat: repeat-x

# UI15 with Bootstrap
$navbar-default-bg: #f8f8f8
$navbar-default-color: #777777
$brand-warning: #df8a13
$brand-primary: #428bca
$brand-info: #5bc0de
$jumbotron-bg-color: #428bca
$panel-bg: #ffffff
$state-success-text: ${g_darken($color-positive, 25%)}
$state-success-bg: ${g_lighten($color-green-lighter, 20%)}
$state-success-border: ${g_lighten($color-positive, 30%)}

$state-info-text: ${g_darken($color-info, 25%)}
$state-info-bg: ${g_lighten($color-blue-lighter, 8%)}
$state-info-border: ${g_lighten($color-info, 30%)}

$state-warning-text: ${g_darken($color-warning, 25%)}
$state-warning-bg: ${g_lighten($color-orange-lighter, 10%)}
$state-warning-border: ${g_lighten($color-warning, 30%)}

$state-danger-text: ${g_darken($color-negative, 25%)}
$state-danger-bg: ${g_lighten($color-red-lighter, 8%)}
$state-danger-border: ${g_lighten($color-negative, 30%)}
```

**Default CSS styles for UI11**

```
base.color: rgb(102, 153, 204)
base.font-family: arial
base.font-size: 10pt
# application font size
menu.font-size: smaller

mobile.font-family: ${base.font-family}
```
mobile.font-size: \{base.font-size\}

mobile.touch.font-family: \{base.font-family\}
mobile.touch.font-size: 13pt

button.action.text.color: \{base.text.color\}
button.action.background.color: \{base.color\}
button.action.border.color: \#666666
button.action.border.width: 1px

button.login.text.color: \{base.text.color\}
button.login.background.color: \{base.color\}
button.login.border.color: \#666666
button.login.border.width: 1px

banner.title.color: \{base.color\}
banner.description.color: \#808080
text.caption.color: \{base.color\}

# Banner branding styles
banner.background.attachment: scroll
banner.background.color: \#FFF
banner.background.image:url("../images/s.gifx")
banner.background.position: center bottom
banner.background.repeat: repeat-x

body.background.color: \{base.background.color\}

table.background.color: \{base.background.color\}
table.column.head.background.color: \{base.color\}
table.column.head.text.color: \{base.text.color\}

button.header.text.color: \{base.text.color\}
button.header.background.color: \{table.column.head.background.color\}
button.header.border.color: white
button.header.border.width: 1px

label.text.color: black
label.background.color: \#dddddd
label.mandatory.text.color: \{base.text.color\}
label.mandatory.background.color: \#cc0000
label.mandatory_populated.text.color: \{base.text.color\}
label.mandatory_populated.background.color: \#cc7777
label.read_only.background.color: orange
label.changed.background.color: \#00cc00
label.foreign.text.color: black
label.foreign.background.color: silver
label.error.text.color: black
label.error.background.color: khaki
label.debug.text.color: black
label.debug.background.color: khaki
label.debug.success.text.color: black
label.debug.success.background.color: \#99FF33
label.debug.failure.text.color: black
label.debug.failure.background.color: \#FF9933

list.base.text.color: black

list.row.even.background.color: \#EEEEEE
list.row.highlight.background.color: \#EEE

list.row.even.text.color: \{list.base.text.color\}
list.row.odd.background.color: \{base.background.color\}
nav_header.background.color: ${base.color}
nav_header.text.color: ${base.text.color}
nav_header.margin-left: 16px

drag.section.border: #d5d5d5
drag.section.border.dark: #999999
drag.section.selected.border: #aabbdd
drag.section.dragging.border: #222222
drag.section.hilight.border: #00ff00
drag.section.hilight.neighbors.border: #ff9900
drag.section.top.background.color: #eeeeee
drag.section.header.background.color: #eeeeee
drag.section.header.active.background.color: #ccddff

table.white-space: normal
table.cell.padding: 2px

input.disabled.text.color: #444444
input.disabled.background.color:#f6f6f6

input.ref_dynamic.border.color: #00CC00
input.ref_invalid.border.color: #DC143C

textsearch.kb.background.color: #f0f7f9
textsearch.catalog.background.color: #ffffff

questionset.text.width: 250px
questionset.reference.width: 157px

user_image.max.height: 250px
user_image.max.width: 250px

# All the properties needs to be moved in a new Table_UI_Properties file

# Tablet header and footer background
tablet.gradient.start: #666666
tablet.gradient.end: #111111

# Tablet header footer text color
tablet.headerfooter.text.color: #ffffff

script.editor.width: 80%

#mobile
m.label.color: #666
m.separator.color: #DDD
m.border-box: -moz-box-sizing: border-box; box-sizing: border-box;
m.context_menu.background: background: none repeat scroll 0 0 #FFFFFF;
m.context_menu.border: border: 1px solid darkgrey; border: 1px solid rgba(0, 0, 0, 0.2); border-radius: 4px;
m.context_menu.element.font: font-weight: bold; color: #000;
m.context_menu.element.padding: padding: 5px;
m.one-line-clipped: white-space: nowrap; text-overflow: ellipsis; overflow: hidden;

m.message.background.error: #F2DEDE
m.message.error: background-color: ${m.message.background.error}; border-color: #EED3D7; color: #B94A48;
m.message.arrow.error: ${m.message.background.error}

m.message.background.info: #D9EDF7
m.message.info: background-color: ${m.message.background.info}; border-color: #BCE8F1; color: #3A87AD;
UI properties

You can customize the following theming properties by navigating to **System Properties > UI Properties**.

- Icons used in the activity formatter
- Background colors for Additional Comments and Work Notes
- Button placement on forms
- Icons used in the Task Activity formatter
- Background colors for Incident Additional Comments and Work Notes

Create a company profile

To customize the ServiceNow® instance for your company, you can enter information such as contact phone numbers, street address, and additional notes. You can also customize the company logo and banner text your end users see at the top of each page.

Role required: admin

Much of the company information that you enter is reference information that administrators can view. All users see the company logo and banner text. To see all company information, verify that you are in the **My Company** view.
1. Navigate to System Properties > My Company.
2. To change the banner text, update the Banner text field.
3. Select the Primary check box to indicate if this company is the primary company.
   If needed, configure the form to add the Primary check box. Designate only one company in your system as the primary company.
4. To upload your company logo, complete one of the following steps for your version of the UI.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI16</td>
<td>Click Click to add or Update beside UI16 Banner Image.</td>
</tr>
<tr>
<td>UI15 or UI11</td>
<td>Click Click to add or Update below the Banner image field.</td>
</tr>
</tbody>
</table>

If you leave the banner image blank for either UI16 or UI15, the system uses the image used in System Properties > Basic Configuration > Banner image [glide.product.image] as the default.
5. Click Choose file and select the file, and then click OK.
   To use an image URL instead of a file on your hard drive, enter the URL in the file upload window.
6. Complete the form with remaining company information.
7. Click Update.
Define the company profile Apple icon

When viewing an instance on an Apple device, users can save a bookmark to the home screen that navigates directly to the instance. The icon that the bookmark uses can be defined in the company profile.
1. Navigate to System Properties > My Company.
2. If necessary, *Configure the form* and add the **Apple Icon** field.

3. Use the **Apple Icon** field to upload the desired image.
   - This should be a 57x57 px image in `.png` format:
   - When users save the bookmark to the home screen, it will display the new icon.
Banner logo link

Properties are available to control the URL and target frame used when clicking the banner logo.

- glide.banner.image.url: URL used when clicking the banner image
- glide.banner.image.url_target: Target frame used when clicking the banner image
  - gsft_main for the main frame
  - _top to replace the current browser window
  - _blank for a new window/tab

Customize the logo in system properties

The banner text and banner image defined for the company that a user is assigned to may be used to override the glide.product.image and glide.product.description.

Role required: admin

The banner text, if specified, is used instead of the glide.product.description property. The banner image, if specified, is used instead of the glide.product.image property.

To update the banner image in the System Properties > System page, you must first upload the image.

Note: Setting a banner image in the company profile overrides the image selected in System Properties > System.

1. To upload the image, complete the following steps.
   a) Navigate to System UI > Images and click New.
   b) Enter a Name for your image.
      Valid names must end in .gif, .png, .jpg, or .bmp.
   c) Click the Click to add link in the Image field, and select and upload the image.
   d) Click Update.

2. To set the uploaded image to the banner image, complete the following steps.
   a) Navigate to System Properties > System.
   b) In the Banner image field, enter the file name of the uploaded image.
   c) Click Save.

Note: The system limits banner images to 50px.

Customize the favicon

Use the glide.product.icon property to change the icon that appears in bookmarks and the browser address bar.

Before you can update the favicon, you need to upload the image into the database.

Role required: admin

1. To upload the image, complete the following steps.
   a) Navigate to System UI > Images and click New.
b) Enter a **Name** for your image. Valid names must end in .gif, .png, .jpg, .ico or .bmp.

c) Click the **Click to add** link in the **Image** field, then select and upload the image.

d) Click **Update**.

2. To set the uploaded image to the favicon, complete the following steps.

   a) Navigate to **System Properties > System**.

   b) In the **Icon image displayed in the bookmarks and browser address bar** field, enter the file name of the uploaded image.

   c) Click **Save**.

The favicon appears in the bookmarks and browser address bar for the platform or the Content Management System (CMS).

**Examples of how to modify the banner**

There are various ways that you can modify the banner on your instances.

**Label multiple instances differently**

- On all instances, leave glide.product.description <blank>
- On PROD instance, set glide.product.name to <My Company>
- On DEV instance, set glide.product.name to <My Company - DEV>

**Control the window title**

Since the window title is composed of glide.product.name and glide.product.description, the following gives you complete control over the banner and the window title.

- glide.product.name set to <Window Title that you want>
- glide.product.name.style set to <display: none>
- glide.product.description set to <blank>

The company record **Banner Image** and **Banner Text** are used to set the banner image and text since the company banner text is not used in the window title.

**Use HTML in the banner text**

- glide.product.name set to <My Company>
- glide.product.name.style set to <display: none>
- glide.product.description set to <Some text <a href="some_url">Click here</a>>

**Position banner text over the banner image**

- glide.product.name set to <My Company>
Apply a background image to the banner

Often corporate guidelines require more in-depth branding of the ServiceNow interface. Using the tiling technique, you can modify your logo image to have a transparent background, so that you can apply another image to the banner beneath the logo.

Upload an image to the image manager and then create a new property named css.banner.background.image, so that you can call the image. The value should look like this.

- css.banner.background.image value url('./images/MasterBG.jpgx')

Write CSS rules that apply this change

```css
/** BACKGROUND - These properties allow you to add a background tile to the header of the instance *****************************/
TD.bannerLeft, TD.bannerCenter, TD.bannerRight, TR#banner_row{
    background-color: ${banner.background.color};
    background-image: ${banner.background.image};
    background-position: ${banner.background.position};
    background-repeat: ${banner.background.repeat};
}
```

Note: The banner image and banner text associated with the primary company or specific companies may be used instead of the glide.product.image and glide.product.description properties to get the same effect.

Modify the banner

The banner is displayed at the top of the page and is rendered using certain system properties.

- glide.product.image
- <div>glide.product.name</div>
- <div>glide.product.description</div>

The DIV that contains the glide.product.name property is only shown if the property contains a value, otherwise, it is not used when rendering the banner. Also, the property glide.banner.image.title controls the tool tip that appears when the cursor is over the banner.

Note: The properties are overridden by the My Company record.
These properties are used to set the window title as follows:

\[ \text{glide.product.name} \text{ glide.product.description} \]

If \text{glide.product.name} is blank, then the ServiceNow name is used as the product name for the window title.

**My Company**

The banner text and banner image defined for the Company that a user is assigned to may be used to override the \text{glide.product.image} and \text{glide.product.description} properties. The banner text, if specified, is used instead of the \text{glide.product.description} property. The banner image, if specified, is used instead of the \text{glide.product.image} property.

**Style the banner**

The \text{glide.product.description} property and the company banner text field may contain HTML if necessary to allow links, color, etc. to be shown in the banner.

**Role required:** admin

Use the \text{glide.product.name.style} property to set CSS properties for the DIV used to display the product name in the banner. Often, this style is set to \text{display: none} so that the product name is not displayed, but is still used as part of the window title.

Use the \text{glide.product.description.style} property to set CSS properties for the DIV used to display the product description in the banner, or company banner text.

You can also use CSS properties to change the banner background color by completing the following steps.

1. Navigate to **System Properties > CSS**.
2. In the **Banner and list caption background color** field, enter the color you want to apply to the banner.
3. Click **Save**.

**Comparing field values**

Field comparison allows users to evaluate equality between fields on the same table or on related tables by using operators in the condition builder.

Additionally, you can compare date type fields to determine if the values are within a certain range of each other. For example, you can compare the planned start date of a task to the actual start date. You can use field comparisons in a list view and to generate reports.
The primary function of field comparison is evaluating whether two fields on a record or related record have identical values. For example, you can create a filter to display all incidents whose **Caller** field references the same user record as the **Closed by** field.

**Available operators**

Field comparison provides several filter operators for comparing field values.

- **[is same]** evaluates to true if two field values are the same.
- **[is different]** evaluates to true if two field values are not the same.
- **[is less than]** evaluates to true if two date values are within a user-defined range of each other.
- **[is more than]** evaluates to true if two date values are not within a user-defined range of each other.

**Comparing choice list values**

Field comparison compares the dictionary **Value** field of a choice list, not the **Label** value. For example, a high priority incident has a priority value of 1, not 1 - High. To find the **Value** of a choice, right-click the field label and select **Configure Dictionary**. The **Choices** related list shows the **Value** for each choice.

**Comparing empty fields**

By default, most filter operations do not return empty fields in their result set. To include records with empty fields in a result set, add a filter condition for the left operand with an operator of **[is empty]**. For more information, see **Filtering on empty fields** on page 158.

**Using operators and operands**

When comparing fields, the two fields being compared are called the left operand and right operand.

The type of field selected in the left operand, and the operator selected, determine which right operands are available. When using the **[is same]** or **[is different]** operators, the right operand choices include only fields of the same field type, such as **String** or **Integer**, as the left operand.

For example, a filter on the Task [task] table with a left operand of **[Opened by]** and an operator of **[is same]**, only allows you to choose user reference fields, such as the **[Closed by]** field, in the right operand.

The **[is more than]** and **[is less than]** operators are only available with date type fields.

This example filter returns records where the user who opened the record also closed the record.

![Figure 109: Example filter](image-url)
Comparable fields

Field comparison supports the comparison of several fields. The types of fields that are comparable include the following.

- String, choice, integer, and boolean fields: return true if both values match.
- Reference fields: return true if both reference fields refer to the same record.
- Date and time fields: can match date values based on hour, day, week, month, quarter, or year. Additionally, you can evaluate whether two dates fall within a certain range of each other.

**Note:** Field comparison does not support journal, keyword, script, duration, list, or HTML fields.

Comparing dates

Field comparison allows you to evaluate equality between two components of the date, such as the hour of the day or the date within a month. Additionally, you can evaluate if two dates fall within a certain range of each other.

Date comparisons are based on the active user's time zone. For example, as a user in the PST time zone, a filter of [Created] [is same] [Day] as [Closed] evaluates true for an incident created at 6 AM PST and closed at 3:00 PM PST as both times are within the same day. However, a user in the GMT time zone would not see this same incident record in response to the example filter; in GMT those times do not fall on the same date.

Evaluate equality between date values

Date comparison evaluates a match for one of several date increments.

You can specify granularity to the hour, day, week, month, quarter, or year. For example, the filter [Created on] [is same] [Week] as [Closed] returns records that were closed in the same week that they were opened. When you run this type of filter, be sure to specify the year in an additional filter condition of [Created] [is same] [Year] as [Closed] to eliminate records that were created a year before they were closed.

![Figure 110: Date field comparison](image)

1. In the condition builder, create a new condition.
2. Select a date field from the left operand choice list.
3. Select [is same] or [is different] from the operator choice list.
   - An additional choice list appears.
4. Select the granularity from the date range choice list.

<table>
<thead>
<tr>
<th>Date Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hour</td>
<td>Filters on the hour of the day.</td>
</tr>
<tr>
<td>Day</td>
<td>Filters on the day of the month, not the date. For example, a Day value of the 1st of August matches with a Day value of the 1st of September.</td>
</tr>
<tr>
<td>Week</td>
<td>Filters on the week of the year. ServiceNow defines a week as Sunday through Saturday for the purposes of Field Comparison.</td>
</tr>
<tr>
<td>Month</td>
<td>Filters on the month of the year. For example, any date within December returns the 12th month of the year.</td>
</tr>
<tr>
<td>Quarter</td>
<td>Filters on the quarter of the year.</td>
</tr>
<tr>
<td>Year</td>
<td>Filters on the year.</td>
</tr>
</tbody>
</table>

5. Select a different date field from the right operand choice list.

6. Click Run.

Evaluating unequal dates within a range

You can compare unequal date field values by evaluating whether two dates fall within a certain range of each other.

For example, you can create a filter of [Created on] [is more than] [3] [Months] [before] Closed to display all records that were created at least three months before they were closed. The numerical value specified, such as the value 3 in the example, must be an integer of no more than two digits. If you need to evaluate a difference larger than 99 units, select a less granular unit of time, such as Months instead of Days.
Figure 111: Unequal dates field comparison

Table 63: Choice field date comparison choices

<table>
<thead>
<tr>
<th>Choice field</th>
<th>Date Comparison Choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator</td>
<td>is more than, is less than</td>
</tr>
<tr>
<td>Unit of time</td>
<td>Hours, Days, Weeks, Months, Quarters, Years</td>
</tr>
<tr>
<td>Relative position of dates</td>
<td>before, after, before or after</td>
</tr>
</tbody>
</table>

Add help to a field label on a form

You can create field-level help in forms by placing help icons in individual fields and converting the field labels to links to external or internal URLs.

Role required: personalize_dictionary

You can link to any type of file, document, or Wiki, and open it in a separate browser window. The following is an example of a form with field label help enabled.
Figure 112: Field label help enabled

1. Open the form.

2. Right-click the label for the field and select **Configure Label**. The Field Label form appears. The fields in the top portion of the form are pre-populated. The **Help** field is not currently used.
3. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hint</td>
<td>Type text to use as a tooltip that appears when a user places the cursor on the field label.</td>
</tr>
<tr>
<td>URL</td>
<td>Enter the URL of the target file to open when the field label or icon is clicked. This can be a complete URL to a file outside the instance, or a relative URL to a target on the instance.</td>
</tr>
<tr>
<td>URL target</td>
<td>(UI11 only) Type _blank to have the help file open in a new tab, or leave the field blank to have the help file replace the instance in the browser. In UI16 and UI15, the help file opens in a new tab regardless of the URL target value.</td>
</tr>
</tbody>
</table>

4. Click **Update**.
Context menus

Context menus in lists of records are right-click menus that provide three different levels of controls, depending on the view from which they are opened. The content and behavior of context menus can be controlled by an administrator.

Certain context menus are configured for administrative users by default. The menu options are for commonly used tables, such as Incident [incident] and Change [change_request]. Other options might appear on menus for other tables.

Menu options presented to users can be controlled with roles.

List title

You can click the title of the list to access options related to viewing and filtering the entire list.

![Context menu list title](image)

Figure 113: Context menu list title

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Changes the view of the list by presenting different information.</td>
</tr>
<tr>
<td>Filters</td>
<td>Changes the values in the filter. Some of the choices are: <strong>None</strong>, <strong>Active</strong>, and <strong>Edit personal filters</strong>.</td>
</tr>
<tr>
<td>Group by</td>
<td>Groups records in a list by the values in a selected field from that table. Any field from the table can be used as a group filter, whether or not it appears in the list.</td>
</tr>
<tr>
<td>Show</td>
<td>Changes the number of rows shown on each page of the list.</td>
</tr>
</tbody>
</table>
List header

You can right-click in the header row of a column to display actions related to that column.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh List</td>
<td>Refreshes the list to show changes immediately.</td>
</tr>
<tr>
<td>Create Favorite</td>
<td>Adds a link to this list to your favorites.</td>
</tr>
</tbody>
</table>

Figure 114: Context menu list header
**Table 65: Context menu list header options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort (a to z)/(z to a)</td>
<td>Sort the selected column in ascending or descending order.</td>
</tr>
<tr>
<td>Show Visual Task Board</td>
<td>Create a visual task board based on the current list.</td>
</tr>
<tr>
<td>Group by &lt;column&gt;</td>
<td>Group records by the values in the selected column and arrange the groups alphabetically.</td>
</tr>
<tr>
<td>Charts</td>
<td>Create bar or pie charts on the fly for the values in the selected column.</td>
</tr>
<tr>
<td>Configure</td>
<td>Configure the list for all users on this instance. This includes the layout, calculations, and controls. Use these controls to access other features such as UI Actions, UI Policies, and the Dictionary.</td>
</tr>
<tr>
<td>Import</td>
<td>Import data from an Excel template file.</td>
</tr>
<tr>
<td>Export</td>
<td>Export all the records in the list using the selected file format.</td>
</tr>
<tr>
<td>Update Selected/All</td>
<td>Perform updates to a single record or to multiple records at once using the appropriate form for that table. See Edit multiple records in a list using the list editor on page 92 for instructions. This option requires the list_updater role.</td>
</tr>
<tr>
<td>Create Application Files</td>
<td>Creates demo data from the current list of records that can be included when you install or update the application on another instance. Used with custom application development.</td>
</tr>
<tr>
<td>Import XML</td>
<td>Import records into this table from an external XML file.</td>
</tr>
</tbody>
</table>

**List row**

You can right-click in a row to see a menu with actions related to the values in that row.
Figure 115: Context menu list row

Table 66: Context menu list row options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Matching</td>
<td>Displays only those records containing values that match the cell selected.</td>
</tr>
<tr>
<td>Filter Out</td>
<td>Filters out records containing values that match the cell selected.</td>
</tr>
<tr>
<td>Copy URL to Clipboard</td>
<td>Copies the URL for the selected record to the clipboard.</td>
</tr>
<tr>
<td>Copy sys_id</td>
<td>Copies the sys_id (unique record identifier) of the record selected to the clipboard.</td>
</tr>
<tr>
<td>Assign Tag</td>
<td>Create new tags or select existing tags to organize records under <strong>Self-ServiceMy Tagged Documents</strong>.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Approve</td>
<td>Approves the request in the selected record(s). This option is only available for tables in which records are subject to approvals.</td>
</tr>
<tr>
<td>Follow on Live Feed</td>
<td>Create or join a record feed for this record.</td>
</tr>
<tr>
<td>Assign to me</td>
<td>Assigns the selected record to the current user. This option is only available for tables in which records can be assigned.</td>
</tr>
<tr>
<td>Show Live Feed</td>
<td>Open the record feed for this record.</td>
</tr>
<tr>
<td>Reject</td>
<td>Rejects the request in the selected record(s). This option is only available for tables in which records are subject to approvals.</td>
</tr>
<tr>
<td>Add to Visual Task Board</td>
<td>Lets you add the record to one of your visual task boards.</td>
</tr>
</tbody>
</table>

Create a context menu option

You can create an option for a context menu.

Role required: admin

1. Navigate to **System UI > UI Context Menus**.
2. Click **New**.
3. Complete the form.

---

**Table 67: Context menu form fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select the table to which this context menu option is attached. The base system menu items are attached to the Global [global] table, which applies the context menu option to all lists for all tables. If you specify a particular table, the option is available only on context menus in lists from that table.</td>
</tr>
</tbody>
</table>

**Note:** The list shows only tables and database views that are in the same scope as the context menu.

| Menu   | Select the menu in which this option appears.  
|--------|-----------------------------------------------|
|        | • List title menu  
|        | • List header menu  
|        | • List row menu  

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Select the type of menu option to create:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Action</strong>: A menu option that performs an immediate action.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Menu</strong>: Creates a parent menu that can display a submenu.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Separator</strong>: Draws a line between groups of options on a menu. Menus do not display separator lines adjacent to one another or at the bottom of a menu. If a condition that removes options for a role forces two separators together, one of the separators is removed from the view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Label</strong>: Create an unlinked label for a menu or section of a menu.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Dynamic actions</strong>: Menu options dynamically created, such as the available views or user filters that can only be generated at the time the list is displayed.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Enter the label for the action as it will appear in the menu.</td>
</tr>
<tr>
<td><strong>Parent</strong></td>
<td>If this action is part of a submenu, type the name of the parent menu item. For example, in the base system, <strong>Configure</strong> is a parent.</td>
</tr>
<tr>
<td><strong>Order</strong></td>
<td>Assign an order number to this item, menu, or separator to specify where in the menu it appears.</td>
</tr>
<tr>
<td><strong>Active</strong></td>
<td>Enable or disable this context menu item. Only active items are shown in the context menu.</td>
</tr>
<tr>
<td><strong>Run onShow script</strong></td>
<td>Select this check box to display the onShow script field.</td>
</tr>
<tr>
<td><strong>Condition</strong></td>
<td>Create the conditions under which this menu option appears. For example, define the role that has permission to see this item.</td>
</tr>
<tr>
<td><strong>Action script</strong></td>
<td>The action script is the code that runs in response to the menu item that is selected.</td>
</tr>
<tr>
<td><strong>Dynamic actions script</strong></td>
<td>The dynamic actions script builds the dynamic items that appear on the menu, such as filters or views.</td>
</tr>
<tr>
<td><strong>onShow script</strong></td>
<td>The onShow script contains a script that runs before the menu is displayed that determines the items to present based on previous settings.</td>
</tr>
</tbody>
</table>
```javascript
function showDatabaseView(query) {
    var url = new GlideURL('sys_db_view.do?g_list-getReferringURL()');
    url.addParam('sysparm_referring_url', query); // use the query string here
    url.addParam('sysparm_domain', 'false');
    window.location = url.getString();
}

function buildContextActions() {
    var queryGen = new GlideCollectionQueryCalculator();
    var query = queryGen.buildQueryClause('g_tableName', 'name');
    g_contextMenu.addAction('dictionary', 'showDatabaseView', query + ' '); // Query is 'query'
}
```
4. Click **Submit**.

**Action script**

The action script is the code that runs in response to the menu item being selected.

This is a client-side script that runs in the user’s browser. The following JavaScript variables are available to the Action script when it is executed.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_list</td>
<td>GlideList2 against which the script runs.</td>
</tr>
<tr>
<td>g.FieldName</td>
<td>Name of the field against which the context menu runs.</td>
</tr>
<tr>
<td>g.FieldLabel</td>
<td>Label of the field against which the context menu runs.</td>
</tr>
<tr>
<td>g.sysId</td>
<td>The sys_id of the row or form against which the script runs.</td>
</tr>
</tbody>
</table>

The base system uses the following code in an action script to refresh the platform view.

```javascript
g_list.refresh(1);
```

Another example is the use of these variables in a list header menu to sort a list by the selected field in descending order (z to a).

```javascript
g_list.sortDescending(g.FieldName);
```

**Dynamic actions script**

The dynamic actions script builds the dynamic items that appear on the menu, such as filters or views.

The following JavaScript variables are available to the dynamic actions script when it is executed.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_tableName</td>
<td>Name of the current table.</td>
</tr>
<tr>
<td>g_listId</td>
<td>ID of the list for which the context menu is built.</td>
</tr>
<tr>
<td>g_itemName</td>
<td>Name defined in the UI context menu record.</td>
</tr>
<tr>
<td>g_itemOrder</td>
<td>Order defined in the UI context menu record. Use this variable to pass the value of the <strong>Order</strong> field to the dynamic actions script.</td>
</tr>
<tr>
<td>gContextMenu.addAction(item_id, label, script_string, order)</td>
<td>Add options to the context menu and select the order in which they appear.</td>
</tr>
</tbody>
</table>
The following example displays a list title menu item that controls the number of records per page in the list view.

```
g_contextMenu.addAction('50', g_itemName, 'showRowsPerPage("50");', g_itemOrder);
```

**Note:** The action script for this item must define the `showRowsPerPage` function so that when selecting this menu item, that function is called with an argument of 50.

### onShow script

The `onShow` script contains a script that runs before the menu is displayed that determines the items to be present based on current information about the menu being displayed.

The `onShow` script is typically used to change the menu items on the list header menu based on the current field column. The following JavaScript variables are available to the `onShow` script when it is executed:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>g_menu</code></td>
<td>Context menu to be displayed.</td>
</tr>
<tr>
<td><code>g_item</code></td>
<td>Current context menu item.</td>
</tr>
<tr>
<td><code>g_list</code></td>
<td>GlideList2 against which the script runs.</td>
</tr>
<tr>
<td><code>g_fieldName</code></td>
<td>Name of the field against which the context menu runs.</td>
</tr>
<tr>
<td><code>g_fieldLabel</code></td>
<td>Label of the field against which the context menu runs.</td>
</tr>
<tr>
<td><code>g_sysId</code></td>
<td>The sys_id of the row or form against which the script runs.</td>
</tr>
</tbody>
</table>

An example of an `onShow` script is one that determines when to enable or disable the **Ungroup** option in a list header menu based on whether the list is currently grouped or not.

```
if (g_list.getGroupBy()) {
    // list is grouped so enable the Ungroup menu item
    g_menu.setEnabled(g_item);
} else {
    // list is not grouped, so disable the Ungroup menu item
    g_menu.setDisabled(g_item);
}
```

### Administer the ServiceNow Platform

The primary administrator role (admin) has access to all system features, functions, and data, regardless of security constraints. Additional administrator roles manage specific applications.

ServiceNow administrators can configure the system in many different ways to address an organization’s business needs. They can install and set up any number of the process applications that are provided with ServiceNow and also create custom applications. Applications can be made available to selected
users by way of roles and other access controls. The system may also be configured to send and receive
notifications as a way to create and track service requests.

Service administration

Configure settings for services that support business applications or the platform.

Assessments

Use assessments to evaluate, score, and rank records from any table in the system.
You can assess projects that executives want to evaluate for effective management, or rate and compare
a vendor's goods and services. Use assessments to send custom questionnaires to selected users or write
scripts that query the database directly. You can then compare assessment results for the records using
unique graphical views designed to highlight key performance information. Share an assessment between
ServiceNow instances by exporting and importing the assessment as an XML file.

The Assessments plugin is enabled by default.

Domain separation

Assessments support domain separation. All assessable records be in the same domain as the elements
required to perform the assessment. A metric type in the global domain can only assess records in the
global domain. The categories, metrics, and users associated with that type must all be in the same
domain. To assess records in a different domain, the type, categories, and metrics must be created in that
domain.

Installed with Assessments

Several types of components are installed with Assessments.
Demo data is available for assessments and surveys. Vendor Performance offers an additional set of
vendor assessment demo data.

Tables installed with assessments

Tables

Assessments adds the following tables.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description: Assessments</th>
<th>Description: Surveys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessable Record [asmt_assessable_record]</td>
<td>Stores all assessable records.</td>
<td>For system use only.</td>
</tr>
<tr>
<td>Assessment Category Result</td>
<td>Stores all category results.</td>
<td>For system use only.</td>
</tr>
<tr>
<td>[asmt_category_result]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>Description: Assessments</td>
<td>Description: Surveys</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Assessment Category User</td>
<td>Stores associations between users and metric categories they can assess. A user that is associated to a category is a category user.</td>
<td>Stores associations between users and surveys. A user that is associated to a survey is a survey user and is automatically associated to all the survey’s categories.</td>
</tr>
<tr>
<td>Assessment Group</td>
<td>Stores all assessment groups.</td>
<td>For system use only.</td>
</tr>
<tr>
<td>Assessment Instance</td>
<td>Stores all assessment instances.</td>
<td>Stores all survey instances.</td>
</tr>
<tr>
<td>Assessment Instance Question</td>
<td>Stores all assessment instance questions, each of which represents one question on one assessment instance.</td>
<td>Stores all survey instance questions, each of which represents one question on one survey instance.</td>
</tr>
<tr>
<td>Assessment Metric</td>
<td>Stores all metrics.</td>
<td>Stores all survey questions.</td>
</tr>
<tr>
<td>Assessment Metric Definition</td>
<td>Stores all metric definitions.</td>
<td>Stores all answer options for survey questions.</td>
</tr>
<tr>
<td>Assessment Metric Template</td>
<td>Stores all metric templates.</td>
<td>Stores all question templates.</td>
</tr>
<tr>
<td>Assessment Metric Type</td>
<td>Stores all metric types.</td>
<td>Stores all survey definitions.</td>
</tr>
<tr>
<td>Assessment Stakeholders</td>
<td>Stores associations between assessable records and category users. A category user that is associated to an assessable record is a stakeholder.</td>
<td>Not used for surveys.</td>
</tr>
<tr>
<td>Assessment Template Definition</td>
<td>Stores all metric template definitions.</td>
<td>Stores all question template definitions.</td>
</tr>
<tr>
<td>Assessment X Category Matrix</td>
<td>Allows users to specify metric categories as the X axis of a decision matrix.</td>
<td>Not used for surveys.</td>
</tr>
<tr>
<td>Assessment Y Category Matrix</td>
<td>Allows users to specify metric categories as the Y axis of a decision matrix.</td>
<td>Not used for surveys.</td>
</tr>
<tr>
<td>Bubble Chart</td>
<td>Stores all bubble chart definitions.</td>
<td>Not used for surveys.</td>
</tr>
<tr>
<td>Table</td>
<td>Description: Assessments</td>
<td>Description: Surveys</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Category Assessable Records</td>
<td>Stores associations between assessable records and metric categories.</td>
<td>For system use only.</td>
</tr>
<tr>
<td>[asmt_m2m_category_assessment]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision Matrix</td>
<td>Stores all decision matrixes.</td>
<td>Not used for surveys.</td>
</tr>
<tr>
<td>[asmt_decision_matrix]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metric Category</td>
<td>Stores all metric categories.</td>
<td>Stores all survey categories.</td>
</tr>
<tr>
<td>[asmt_metric_category]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metric Result</td>
<td>Stores all metric results.</td>
<td>Stores all survey responses.</td>
</tr>
<tr>
<td>[asmt_metric_result]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signature</td>
<td>Stores all signature records for assessments and [[Attestations</td>
<td>GRC attestations]].</td>
</tr>
<tr>
<td>[asmt_signature]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger Condition</td>
<td>Stores all assessment trigger conditions.</td>
<td>Stores all survey trigger conditions.</td>
</tr>
<tr>
<td>[asmt_condition]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Properties installed with assessments*

**Properties**

Assessments adds the following properties.

**Table 72: Properties for assessments**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.snc.assessment.signature_authentication</td>
<td>Require authentication for user signature. When Yes is selected, this property requires credentials for a full name signature.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> True/False</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location:</strong></td>
</tr>
<tr>
<td></td>
<td>• Assessments &gt; Admin &gt; Assessment Properties</td>
</tr>
<tr>
<td></td>
<td>• Survey Management &gt; Administration &gt; Properties</td>
</tr>
<tr>
<td></td>
<td>• <strong>Learn more:</strong> Signatures</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>css.assessment.caption.background.color</td>
<td>Sets the background color of the caption on assessment and survey questionnaires.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: color</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #eee</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #eee</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #eee</td>
</tr>
<tr>
<td>css.assessment.caption.font.color</td>
<td>Sets the font color of the caption text on assessment and survey questionnaires.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: color</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #ffffff</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #ffffff</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #ffffff</td>
</tr>
<tr>
<td>css.assessment.question.header.background.color</td>
<td>Sets the background color of question headers on assessment and survey questionnaires.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: color</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #767676</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #767676</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #767676</td>
</tr>
</tbody>
</table>

**User roles installed with assessments**

**Roles**

Assessments adds the following roles.
Table 73: Roles for assessments

<table>
<thead>
<tr>
<th>Role title [name]</th>
<th>Description</th>
<th>Contains roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>assessment administrator [assessment_admin]</td>
<td>Can administer the Assessments application. Can access all the modules of the Assessments application.</td>
<td>• None</td>
</tr>
</tbody>
</table>

**Note:** The assessment_admin role is contained by the itil_admin role. It is also contained by the survey_admin role.

---

**Script includes installed with assessments**

**Script includes**

Assessments adds the following script includes.

Table 74: Script includes for assessments

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssessmentUtils</td>
<td>Core code for assessment functionality.</td>
</tr>
<tr>
<td>AssessmentUtilsAJAX</td>
<td>AJAX based code for displaying scorecard data.</td>
</tr>
<tr>
<td>MigrateSurveyToAssessment</td>
<td>Code for migrating legacy surveys to assessment surveys.</td>
</tr>
<tr>
<td>RatingUtilsAJAX</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>SurveyUtils</td>
<td>Core code for survey functionality.</td>
</tr>
</tbody>
</table>

---

**Client scripts installed with assessments**

**Client scripts**

Assessments adds the following client scripts.

Table 75: Client scripts for assessments

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate Assessable Record Field choice list</td>
<td>Trigger Condition [asmt_condition]</td>
<td>Limits the Assessable Record Field choices to those that are compatible with the selected assessment.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Check survey schedule period</td>
<td>Trigger Condition [asmt_condition]</td>
<td>Checks the selected survey definition's Schedule period. If Schedule period is set to Only Once, this client script displays a warning message that each user can only take the selected survey once.</td>
</tr>
<tr>
<td>Clear display_when when depends changed</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Shows the Display when field when the Depends on field is set. Clears the Display when field when the Depends on field changes and the options need to change.</td>
</tr>
<tr>
<td>Conditionally Set Type for Surveys</td>
<td>Group [sys_user_group]</td>
<td>Sets the user group Type to survey if you create a new group from the User Groups module.</td>
</tr>
<tr>
<td>Data type -- Hide choices for surveys</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Ensures only Data type options that are valid for surveys are available on the Survey Question form.</td>
</tr>
<tr>
<td>Data type -- Show/Hide Not Applicable</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Hides the Allow not applicable field and sets its value to false if the metric method is Assessment and the data type is Percentage, Checkbox, Date, Date/Time, Number, or String.</td>
</tr>
<tr>
<td>Datatype defaults to String for surveys</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Sets the default Data type field value to String on the Survey Question form.</td>
</tr>
<tr>
<td>Fetch Min/Max</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Updates the Min and Max fields to match the least and greatest metric template definition values if an assessment administrator selects or changes the Template value.</td>
</tr>
<tr>
<td>Hide Assessable Record Field</td>
<td>Trigger Condition [asmt_condition]</td>
<td>Hides the Assessable Record field unless the evaluation method of the metric type selected in the Assessment field is Assessment.</td>
</tr>
<tr>
<td>Hide assessment if empty</td>
<td>Survey [survey_master]</td>
<td>Hides the Assessment field on the Survey form unless it contains a value. The system populates the Assessment field when you migrate a survey.</td>
</tr>
<tr>
<td>Hide Survey Instance Trigger ID if Empty</td>
<td>Assessment Instance [asmt_assessment_instance]</td>
<td>Hides the Trigger ID field and makes it read-only unless it contains a value.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hide user lists</td>
<td>Assessable Record [asmt_assessable_record]</td>
<td>Hides the Category users related list if the associated metric type schedule type is On demand.</td>
</tr>
<tr>
<td>Hide user related lists</td>
<td>Metric Category [asmt_metric_category]</td>
<td>Hides the User related list if the associated metric type schedule type is On demand and the Evaluation method is not Survey.</td>
</tr>
<tr>
<td>Hide value or String value</td>
<td>Assessment Instance Question [asmt_assessment_instance_question]</td>
<td>Hides the Value field and shows the String value field if the associated metric data type is Date, Date/Time, or String. Does the opposite if the data type is anything else.</td>
</tr>
<tr>
<td>Live feed deletion warning</td>
<td>Assessable Record [asmt_assessable_record]</td>
<td>Displays a warning message if an assessment administrator clears the Live feed check box.</td>
</tr>
<tr>
<td>Method -- Show/Hide Not Applicable</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Hides the Mandatory and Allow not applicable fields and sets both values to false if the metric method is Script. Shows the Mandatory field if the metric method is Assessment. Shows the Allow not applicable field if the method is Assessment except when the data type is Percentage, Checkbox, Date, Date/Time, Number, or String.</td>
</tr>
<tr>
<td>Min/Max Control</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Makes the Min and Max fields mandatory except when the data type is Template.</td>
</tr>
<tr>
<td>Min/Max Read-only</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Makes the Min and Max fields read-only when the data type is Choice or Likert Scale.</td>
</tr>
<tr>
<td>Populate Groups</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Populates the available values for the Filter field with fields from the selected metric type table.</td>
</tr>
<tr>
<td>Read only Type</td>
<td>Metric Category [asmt_metric_category]</td>
<td>Makes the Type field read-only when it contains a value.</td>
</tr>
<tr>
<td>Reload Decision Matrix Filter Columns</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Populates the available choices for the Default matrix filter as appropriate if the Filter field value changes.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Reload default filters</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Populates the available choices for the Default filter as appropriate if the Display all filters value changes.</td>
</tr>
<tr>
<td>Set scale factor</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>Set table field</td>
<td>Metric Category [asmt_metric_category]</td>
<td>Sets the category Table value to that of the metric type.</td>
</tr>
<tr>
<td>Show and Hide Scheduled Job</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Hides the Scheduled job field if the Schedule type is On demand and shows the Scheduled job field if the Schedule type is Scheduled.</td>
</tr>
<tr>
<td>Show and Hide Scheduled Job</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Shows the Scheduled job field on the Survey Definition form if the Schedule period is Daily, Weekly, Monthly, or Yearly.</td>
</tr>
<tr>
<td>Survey View</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Displays the Assessment Metric Definitions related list if the data type is Choice or Likert Scale.</td>
</tr>
<tr>
<td>Toggle Metric Definitions (Load)</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Displays the Assessment Metric Definitions related list if the data type is Choice or Likert Scale. Sets the Method field to the appropriate value if the data type is compatible with one method only.</td>
</tr>
<tr>
<td>Toggle Metric Definitions (Update)</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Displays the Assessment Metric Definitions related list if the data type is Choice or Likert Scale. Sets the Method field to the appropriate value if the data type is compatible with one method only.</td>
</tr>
<tr>
<td>Update Min And Max Default Values</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Sets reasonable default values for the Min and Max fields, based on the data type.</td>
</tr>
<tr>
<td>Validate Probability</td>
<td>Survey Trigger Condition [asmt_condition]</td>
<td>Ensures that the Probability (%) value is a whole number between 1 and 100.</td>
</tr>
<tr>
<td>Verify Max</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Verifies that the Max value is greater than the Min value and greater than or equal to zero if the metric method is Assessment.</td>
</tr>
<tr>
<td>Verify Min</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Verifies that the Min value is less than the Max value and greater than or equal to zero if the metric method is Assessment.</td>
</tr>
<tr>
<td>Verify Value</td>
<td>Assessment Metric Definition [asmt_metric_definition]</td>
<td>Verifies that the Value is greater than or equal to zero.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Verify Value</td>
<td>Assessment Template Definition</td>
<td>Verifies that the Value is greater than or equal to zero.</td>
</tr>
<tr>
<td></td>
<td>[asmt_template_definition]</td>
<td></td>
</tr>
</tbody>
</table>

**Business rules installed with assessments**

**Business rules**

Assessments adds the following business rules.

**Table 76: Business rules for assessments**

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessable domain matches type</td>
<td>Assessable Record [asmt_assessable_record]</td>
<td>Ensures that all new assessable records are in the same domain as the metric type. This is a special requirement of assessment domain separation, that all records must be in the same domain.</td>
</tr>
<tr>
<td>Auto stakeholder creation</td>
<td>Assessment Category User [asmt_m2m_category_user]</td>
<td>Assigns new category users as stakeholders automatically for assessable records if the associated category has the Create stakeholders check box selected.</td>
</tr>
</tbody>
</table>
| Auto stakeholder creation              | Category Assessable Records [asmt_m2m_category_assessment] | Creates stakeholders from all of a category’s category users automatically for new assessable records if:  
  • The Create stakeholders check box is selected for the metric category.  
  • The same metric category is associated to the new assessable record. |
<p>| Auto stakeholder creation              | Metric Category [asmt_metric_category]     | Assigns all of a category's category users as stakeholders for each assessable record associated to the category when the Create stakeholders check box is selected for the category. |
| Calculate category max weight          | Assessment Metric [asmt_metric]            | Calculates the sum of all metric weights in a category.                                                                                                          |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel notification workflow</td>
<td>Assessment Instance [asmt_assessment_instance]</td>
<td>Cancels the workflow that generates assessment email notifications when an assessment instance is deleted or changes state to Complete or Canceled.</td>
</tr>
<tr>
<td>Category domain matches type</td>
<td>Metric Category [asmt_metric_category]</td>
<td>Ensures that all new categories are in the same domain as the metric type. This is a special requirement of assessment domain separation, that all records must be in the same domain.</td>
</tr>
<tr>
<td>Check Live Feed Groups</td>
<td>Assessable Record [asmt_assessable_record]</td>
<td>Selects the Live feed check box if a live feed group is associated with the source record. If the Live feed check box is selected and no live feed group is associated with the source record, this business rule clears the check box.</td>
</tr>
<tr>
<td>Check only one default</td>
<td>Bubble Chart [asmt_bubble_chart]</td>
<td>Ensures there is only one default bubble chart for a metric type.</td>
</tr>
<tr>
<td>Check only one default</td>
<td>Decision Matrix [asmt_decision_matrix]</td>
<td>Ensures there is only one default decision matrix for a metric type.</td>
</tr>
<tr>
<td>Create actual results</td>
<td>Assessment Instance [asmt_assessment_instance]</td>
<td>Generates assessment and category results from the user responses if a user completes an assessment.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create Business Rule on Remote table</td>
<td>Assessment Metric Type</td>
<td>Generates the following business rules if an assessment administrator creates or updates a metric type:</td>
</tr>
<tr>
<td></td>
<td>[asmt_metric_type]</td>
<td>• Business rule called Auto business rule for Assessments, which monitors the Table selected for the metric type. When someone adds a record that meets the metric type conditions, this business rule generates an assessable record. If someone changes the metric type's Table or conditions, the Create Business Rule on Remote table business rule updates the automatic business rule script to reflect the changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Business rule called Auto deletion rule for Assessments, which prevents users from deleting a record on the metric type's Table if the record is associated to metric or category results. If the record has no associated results and a user deletes it, this business rule deletes any associated assessable records.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Create Business Rule on Remote table business rule generates automatic business rules only for metric types with Evaluation method set to Assessment.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Create Scheduled Job                         | Assessment Metric Type                     | **Create Scheduled Job**

**Description:**

Assessments: Generates a scheduled job for the creation of assessment components if either of these conditions is met:

- Someone creates a new metric type with the schedule type set to Scheduled.
- The schedule type changes from On demand to Scheduled.

Surveys: Generates a scheduled job for the creation of survey components if the schedule period for a survey definition is set to Daily, Weekly, Monthly, or Yearly.

**Create Survey Records**

**Description:**

Generates various records and populates certain fields on those records when someone creates a new survey definition, which is a metric type with Evaluation method set to Survey. More specifically, this business rule generates a survey category and assessable record, both associated to the survey definition.

**Create UI Action on Remote table**

**Description:**

Reserved for future use.

**decision_matrix_axis**

**Description:**

Limits available categories for X- and Y-axes to those that belong to the type if there is a metric type specified on the Decision Matrix form.

**Delete Live Feed Group**

**Description:**

Deletes the assessable record's live feed group, if there is one, when an assessment administrator deletes an assessable record.

**delete related users**

**Description:**

Deletes any stakeholders for the assessable record and category when an assessment administrator disassociates a category from an assessable record.
<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not allow category to change</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Prevents the Category field from being changed if there are any conditional question dependencies related to the current record.</td>
</tr>
<tr>
<td>Do not allow datatype to change</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Prevents the Data type and Template fields from being changed if there are any conditional questions that depend on the current record.</td>
</tr>
<tr>
<td>Ensure Types Match</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Sets the type of a metric to the type of the associated category.</td>
</tr>
<tr>
<td>Ensure types match</td>
<td>Assessment Stakeholders [asmt_m2m_stakeholder]</td>
<td>Ensures users can only create a stakeholder from a category user and assessable record of the same type.</td>
</tr>
<tr>
<td>Ensure types match</td>
<td>Category Assessable Records [asmt_m2m_category_assessment]</td>
<td>Prevents users from associating categories of one type to an assessable record of a different type.</td>
</tr>
</tbody>
</table>
| Evaluate filters | Assessable Record [asmt_assessable_record] | Performs the following after the system generates a new assessable record:  
  - Checks all filter conditions for categories in the type.  
  - Automatically associates the new assessable record to all categories for which it meets the category filter conditions. |
<p>| Generate assessment trigger condition | Trigger Condition [asmt_condition] | Generates a business rule the survey or assessment trigger condition uses to send surveys or assessments. |
| getStakeholders | Global [global] | Ensures that only category users for categories associated with the assessable record are available on the Assessable Record form when users edit the Category users related list. |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link assessable record and category</td>
<td>Metric Category</td>
<td>Sets these hidden fields as follows for a newly created survey category:&lt;br&gt;  • Table: Assessment Metric Type [asmt_metric_type]&lt;br&gt;  • Filter: &lt;sys_id of the survey definition the new survey category is associated with&gt;</td>
</tr>
<tr>
<td>Live Feed Group</td>
<td>Assessable Record</td>
<td>Creates a live feed group for the assessable record, if one does not exist, when the record is saved with the Live feed check box selected. If the check box is then cleared and the record saved, this business rule deletes an existing live feed group.</td>
</tr>
<tr>
<td>Metric domain matches category</td>
<td>Assessment Metric</td>
<td>Ensures that all new metrics are in the same domain as the category to which they belong. This is a special requirement of assessment domain separation, that all records must be in the same domain.</td>
</tr>
<tr>
<td>Notify assessment user</td>
<td>Assessment Instance</td>
<td>Starts a workflow that sends an email notification to the assigned user when an assessment instance changes to the Ready state or the user to which it is assigned changes. The workflow generates other assessment email notifications at later stages.</td>
</tr>
<tr>
<td>Prevent recursive dependencies</td>
<td>Assessment Metric</td>
<td>Prevents the Depends on field from being set in a way that creates a recursive conditional question dependency.</td>
</tr>
<tr>
<td>Publish Survey</td>
<td>Assessment Metric Type</td>
<td>Generates survey instances and assigns them to the associated survey users when the survey definition state changes from Draft to Published.</td>
</tr>
<tr>
<td>Remove auto create Business Rules</td>
<td>Assessment Metric Type</td>
<td>Deletes the automatically created business rule for a metric type when an assessment administrator deletes that metric type.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Remove auto create UI Actions</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>Remove auto created business rule</td>
<td>Survey Trigger Condition [asmt_condition]</td>
<td>Deletes the automatically created business rule when someone deletes the survey trigger condition.</td>
</tr>
<tr>
<td>Remove Scheduled Job</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td><strong>Assessments:</strong> Deletes the scheduled job for a metric type if either of these conditions is met:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Someone deletes the metric type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Someone changes the schedule type from Scheduled to On demand.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Surveys:</strong> Deletes the scheduled job for a survey definition if the schedule period is set to Only Once or No Limit.</td>
</tr>
<tr>
<td>Reset Min/Max for metric</td>
<td>Assessment Metric Definition [asmt_metric_definition]</td>
<td>Updates the Min and Max fields for metrics and survey questions based on the metric definition Value.</td>
</tr>
<tr>
<td>Set Domain for M2M Cat Assessable Recs</td>
<td>Category Assessable Records [asmt_m2m_category_assessment]</td>
<td>Ensures that assessable records are only associated to categories in the same domain. This is a special requirement of assessment domain separation, that all records must be in the same domain.</td>
</tr>
<tr>
<td>Set Domain for M2M Category Users</td>
<td>Assessment Category User [asmt_m2m_category_user]</td>
<td>Ensures that users are only linked to categories in the same domain. This is a special requirement of assessment domain separation, that all records must be in the same domain.</td>
</tr>
<tr>
<td>Set Domain for M2M Stakeholders</td>
<td>Assessment Stakeholders [asmt_m2m_stakeholder]</td>
<td>Ensures that assessable records are only linked to category users in the same domain. This is a special requirement of assessment domain separation, that all records must be in the same domain.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Set scratchpad fields</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Caches field values for use by other scripts that run on the Assessment Metric Type form.</td>
</tr>
<tr>
<td>Store view type</td>
<td>Metric Category [asmt_metric_category]</td>
<td>Ensures that the Type field only allows users to select metric types used for assessments.</td>
</tr>
<tr>
<td>Synchronize category survey</td>
<td>Assessment Category User [asmt_m2m_category_user]</td>
<td>Ensures that the same users are associated with all of a survey's categories.</td>
</tr>
<tr>
<td>Synchronize survey users and</td>
<td>Metric Category [asmt_metric_category]</td>
<td>Adds all users associated with a survey's existing categories to any newly created category for that survey.</td>
</tr>
<tr>
<td>stakeholder.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update Category Count</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Updates the value in the Total metrics field on the Metric Category form when metrics are added to and deleted from the category.</td>
</tr>
<tr>
<td>Update records that match</td>
<td>Metric Category [asmt_metric_category]</td>
<td>Automatically performs these tasks when an assessment administrator edits the Filter field:</td>
</tr>
<tr>
<td>filter</td>
<td></td>
<td>• Associates the category to assessable records that meet the filter conditions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Disassociates the category from assessable records that do not meet the filter conditions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Assessable records related list reflects these changes when the record is saved.</td>
</tr>
<tr>
<td>Update scheduled job on</td>
<td>Assessment Metric Type [asmt_metric_type]</td>
<td>Updates the survey creation scheduled job to reflect schedule period changes.</td>
</tr>
<tr>
<td>schedule change</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validate mandatory and not</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Sets the Mandatory and Allow not applicable fields to false if the metric method is Script. If the metric data type is Percentage, the business rule sets the Allow not applicable field to false.</td>
</tr>
<tr>
<td>applicable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verify min/max</td>
<td>Assessment Metric [asmt_metric]</td>
<td>Ensures that the Min is greater than or equal to zero and less than the Max. Sets the Min to 0 and the Max to 1 if the data type is Checkbox or Yes/No.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Verify value</td>
<td>Assessment Template Definition</td>
<td>Ensures that the metric template definition Value is:</td>
</tr>
<tr>
<td></td>
<td>[asmt_template_definition]</td>
<td>• Greater than or equal to zero.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Different from the Value for all other metric template definitions related to the template.</td>
</tr>
<tr>
<td>Verify value</td>
<td>Assessment Metric Definition</td>
<td>Ensures that users can only enter a metric definition Value greater than or equal to zero. Ensures that the metric definition Value is different</td>
</tr>
<tr>
<td></td>
<td>[asmt_metric_definition]</td>
<td>from the Value for all other metric definitions related to the metric.</td>
</tr>
</tbody>
</table>

**Administrator tasks**

Before you create assessments, assign the assessment_admin role to the appropriate users in your organization and determine your objectives for the assessment.

Decide which sets of records in the system to assess, which themes you are interested in, and which traits to measure. Consider your organization’s options for obtaining the data to compare. If you intend to use assessment questionnaires, consider which people can answer the questions.

*Assessment roles*

The Assessments application requires certain roles to perform assessment tasks. No role is required to take assessment questionnaires that are assigned to you.
### Role Title | Role Name | Description
--- | --- | ---
Assessment administrator | assessment_admin | Assessment administrators set up assessments. They know which records to evaluate, the criteria on which to evaluate the records, and who to assign assessments to.

#### Note: By default, users with the assessment_admin role have limited system rights and might not have access to all source records to assess. When planning assessments, grant additional roles to assessment administrators as needed. For example, to create and manage vendor assessments, the assessment administrator must also have the vendor_manager role, which grants access to the Company table and other relevant tables.

ITIL user | itil | ITIL users perform basic technician operations in the system. In the Assessments application, they have read access to the Assessable Record table.

Administrator | admin | Administrators have access to all aspects of the assessment process. Only administrators can set up assessment schedules.

---

**Assessments: important terms**
Assessment admins use several terms when working with assessments.
### Assessment terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric type</td>
<td>A <em>metric type</em> defines a set of records to evaluate, such as vendors, projects, or employees.</td>
</tr>
<tr>
<td>Assessable record</td>
<td>An <em>assessable record</em> links a record to evaluate, such as the company record for Amazon or the user record for a sales representative, to a metric type, such as vendors or employees. You define sets of assessable records when you create metric types.</td>
</tr>
<tr>
<td>Metric category</td>
<td>A <em>metric category</em> represents a theme for evaluating assessable records. Categories contain one or more individual metrics, which define specific traits or values that comprise the theme. Examples of categories include overall vendor performance or quality of delivery services. You can also set filter conditions that control which assessable records to evaluate for the metrics in a category.</td>
</tr>
<tr>
<td>Metric</td>
<td>A <em>metric</em> is a trait or value used to evaluate assessable records. A metric can measure subjective values in an assessment questionnaire, or gather objective values in a database query run by a script. Examples of metrics include perceived courtesy of sales representatives or number of incidents per vendor.</td>
</tr>
<tr>
<td>Category user</td>
<td>A <em>category user</em> is a person who knows about a specific category. One person can be a category user for multiple categories. Examples of category users include a vendor manager who oversees all purchasing operations or a supervisor of a sales team.</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>A <em>stakeholder</em> is a category user who knows about a specific assessable record. Examples of stakeholders include a vendor manager who coordinates purchases from Amazon or a supervisor of a sales team who manages a specific employee.</td>
</tr>
</tbody>
</table>
| Assessment instance   | An *assessment instance* represents one assessment questionnaire assigned to one user. The system generates a new assessment instance for each assigned user when:  
  - The assessment generation scheduled job runs.  
  - An assessment administrator creates an on-demand assessment. |
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scorecard</td>
<td>A scorecard provides a visual illustration of an assessable record's performance, based on assessment results. Use scorecards to view a variety of data summaries for one assessable record and to compare the ratings with other assessable records.</td>
</tr>
<tr>
<td>Decision matrix</td>
<td>A decision matrix is a graph with two axes that plots the assessment results for multiple assessable records. Use decision matrixes to determine the relative standing of assessable records in selected categories.</td>
</tr>
<tr>
<td>Bubble chart</td>
<td>A bubble chart is a graph with three axes that plots the assessment results for multiple assessable records. Use bubble charts to determine the relative standing of assessable records in selected categories, with an emphasis on one category.</td>
</tr>
</tbody>
</table>

**Assessment methods**

The available methods are **Assessment**, for non-scripted metrics, and **Script**, for scripted metrics. Each method serves a different function and can be used with certain data types.

Use the **Method** field to specify how to use the metric.

**Assessment method**

Metrics with the Method set to Assessment are called non-scripted metrics. Use each non-scripted metric to define a question for assessment questionnaires. Non-scripted metrics are useful if you want to obtain subjective data like personal opinions.

You can use the Assessment method with these Data type values:

- Checkbox
- Choice
- Date
- Date/Time
- Likert Scale
- Number
- Percentage
- String
- Template
- Yes/No

**Script method**

Metrics with the Method set to Script are called scripted metrics. Use each scripted metric to define a custom script for database queries. Scripted metrics are useful if your system contains reliable data for the traits you want to evaluate.

You can use the Script method with these Data type values:
Use the Script field to write JavaScript code. By default, the field contains information about available variables and an example, which you can use as the basis of your script or replace entirely:

You must use the following variables in your script:

- **primary**: Input variable used to access the sys_id of the record being assessed.
- **actual_result**: Output variable that contains the actual value for this metric. The system uses this variable to populate the Actual value field on the Metric Result form. For each `actual_result`, you must specify a corresponding `scaled_result` value.
- **scaled_result**: Output variable that contains a numerical scaled value to represent an actual value. The system uses this variable to populate the Scaled value field on the Metric Result form. Ensure the scaled values you specify are between or equal to the Min and Max values for the metric. The Scale definition field determines how the system uses the scaled value. A scale definition of **Low** means smaller numbers are better, such as for a metric that measures the number of incidents for a vendor. **High** means larger numbers are better, such as for a metric that measures user satisfaction on a scale of one to five.

**Script example**

The metric Number of active devices uses the script pictured below. The `primary` variable is used to find CIs that are associated to the vendor record being assessed. The script retrieves the `actual_result`, the number of CIs associated to the vendor, then calculates the correct `scaled_result`. The script uses a series of scaled values, from the **Min** to the **Max** value, to represent actual values. Because the Scale definition is set to **High**, the greatest scaled values are best, meaning a vendor associated to the most CIs scores highest. The system stores the actual and scaled values in a metric result record for the vendor.

© 2017 ServiceNow. All rights reserved.  316
Figure 117: Scripted metric example

**Data types for assessments**
Metric data types have functions that depend on the method that you select.

**Table 77: Assessment data types**

<table>
<thead>
<tr>
<th>Data type</th>
<th>Compatible methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkbox</td>
<td>Assessment</td>
<td>On questionnaires, users select a check box next to a statement or leave it cleared.</td>
</tr>
</tbody>
</table>

Set the **Scale definition** field to **High** if a selected check box equates to a good score.
<table>
<thead>
<tr>
<th>Data type</th>
<th>Compatible methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choice</td>
<td>Assessment</td>
<td>On questionnaires, users select a value from a list of custom answer options. Create a metric definition for each answer option.</td>
</tr>
<tr>
<td>Date</td>
<td>Assessment</td>
<td>On questionnaires, users select a date.</td>
</tr>
<tr>
<td>Data type</td>
<td>Compatible methods</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Assessment</td>
<td>On questionnaires, users select a date and time.</td>
</tr>
<tr>
<td>Duration</td>
<td>Script</td>
<td>When the script runs, the system populates the <strong>Duration value</strong> and <strong>Scaled value</strong> fields on the Metric Result form with the appropriate values from the <code>actual_result</code> and <code>scaled_result</code> variables in the metric script.</td>
</tr>
<tr>
<td>Data type</td>
<td>Compatible methods</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Likert Scale</td>
<td>Assessment</td>
<td>On questionnaires, users select a value from a custom scale of answer options. Create a metric definition for each answer option, which is represented by a radio button on the scale. A Likert scale metric that evaluates an application's ease of use might have the metric definitions <strong>Easy, Average,</strong> and <strong>Difficult.</strong> If you want to reuse a series of answer options for multiple metrics, <em>create a metric template</em> and use the Template data type instead of the Likert Scale data type.</td>
</tr>
<tr>
<td>Number</td>
<td>Assessment, Script</td>
<td><strong>Assessment</strong> On questionnaires, users enter a number. <strong>Script</strong> When the script runs, the system populates the <strong>Actual value</strong> and <strong>Scaled value</strong> fields on the <strong>Metric Result</strong> form with the appropriate values from the <strong>actual_result</strong> and <strong>scaled_result</strong> variables in the metric script.</td>
</tr>
<tr>
<td>Data type</td>
<td>Compatible methods</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Percentage   | Assessment, Script | **Assessment** On questionnaires, users enter a number.  
Approximately what percent of the time are your incidents resolved in a week or less?  
0 %  

**Script** When the script runs, the system populates the **Actual value** and **Scaled value** fields on the Metric Result form with the appropriate values from the `actual_result` and `scaled_result` variables in the metric script.
<table>
<thead>
<tr>
<th>Data type</th>
<th>Compatible methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Assessment</td>
<td>On questionnaires, users enter text. The size of the string field depends on the String option you select:</td>
</tr>
<tr>
<td>Template</td>
<td>Assessment</td>
<td>On questionnaires, users select a value from a predefined series of answer options. There must be at least one template defined to use this data type. For a template, there is a template definition for each answer option, which is represented by a radio button.</td>
</tr>
<tr>
<td>Data type</td>
<td>Compatible methods</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Yes/No</td>
<td>Assessment</td>
<td>On questionnaires, users select Yes or No from a list. Set the Scale definition field to High if selecting Yes equates to a better score.</td>
</tr>
</tbody>
</table>

**Assessment generation**

In the Assessments application, administrators or assessment administrators can trigger the system to generate scheduled assessments or on-demand assessments when all the prerequisite steps are completed.

An assessment administrator must publish the metric type to enable assessment generation.

The system performs these tasks when it generates assessments:

- Creates assessment questionnaires from non-scripted metrics and assigns the questionnaires to users. When users complete their assigned questionnaires, the system uses their responses to calculate assessment results.
- Runs scripted metrics from each category to query the database and calculate assessment results.

Each time the system generates assessments, it creates some or all of the following components. Consider having an administrator set a schedule for recurring data cleanup, as the system can potentially generate a considerable amount of assessment data.

- **Assessment group**
- **Assessment instances**
- **Assessment results**

**Assessment groups**

An assessment group is a container for assessment instances and assessment results generated in a single occurrence.

The system generates an assessment group every time the scheduled job runs or the API is called, when there is at least one assessable record associated to a category in the type. You can find assessment group records in **Assessments > Assessments > Assessment Groups**.

The Assessment Group form displays the group **Number**, the associated **Metric type**, and these related lists:

- **Assessment Instances**: Lists all assessment instances within this group. There may be no records in this related list. The system does not generate assessment instances if there are only scripted metrics for the type.
• **Metric Results**: Lists all *metric results* for this group. There may be no records in this related list initially. The system generates metric results immediately for scripted metrics, but not for non-scripted metrics, which appear as questions on assessments and require user response. The system dynamically updates the records in this list as users complete assessment questionnaires.

• **Assessment Category Results**: Lists all *category results* for this group. There may be no records in this related list initially. The system generates category results immediately if there are only scripted metrics in a category. Otherwise, the system does not calculate category results until a user completes an assessment questionnaire that contains questions from the category.

**Note**: To prevent the loss of important assessment data, you cannot delete an assessment group if it contains any assessment instances, metric results, or category results.

*Create an assessment category*

Assessment categories are used in conjunction with assessment metric types and assessment metrics to generate bubble charts.

Role required: assessment_admin or admin

The Demand Management application comes with the Demand assessment metric type, five assessment metric categories, and a number of assessment metrics. From the Assessment Categories section of the application menu, you can modify existing assessment categories and create new ones.

1. Navigate to Demand > Settings > Assessments.
2. Click Create New to create a new record.

<table>
<thead>
<tr>
<th>Table 78: Default demand management assessment categories</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
</tr>
<tr>
<td>Size</td>
</tr>
<tr>
<td>Strategic Alignment</td>
</tr>
<tr>
<td>Risk</td>
</tr>
<tr>
<td>ROI</td>
</tr>
<tr>
<td>Cost</td>
</tr>
</tbody>
</table>

*Assessment metric categories*

In the Assessments application, a metric category represents a theme for evaluating assessable records in a given metric type.

Each category has a numeric weight value to indicate its importance relative to other categories. Within a category, records called *metrics* are the traits or values used to evaluate assessable records. For example, there are many categories within the Vendor metric type, including Support Rating, which contains metrics that measure the quality of vendors’ customer support services. Assessable records must be associated to categories to be eligible for evaluation. Assessment administrators create categories and manage which assessable records each category is associated to.

*Weight categories and metrics*
When you create a metric category or metric, you must specify a weight, a numeric value that indicates the importance of the category or metric relative to other categories and metrics.

The greater the weight value, the more important the item is. The system uses weight values in assessment result calculations. Consider these recommendations when choosing weight values.

- **Determine a weighting scale and use it consistently.** In general, use a scale with a small numeric range. A standard 1-10 scale is usually appropriate, where 1 is least and 10 most important. Understand that increasing the maximum value of a weighting scale impacts all related result calculations.

- **Use the same weighting scale for metric categories and metrics.** Though there is no limitation, consider using the same or very similar weighting scales for both categories and metrics to make it easier to remember which values to use. A scale of 1-10 is an effective and simple weighting system, but may not be suitable for all organizations.

**Note:** If you do not want to use weighting, set the same weight value for all categories and metrics. The default weight value is 10.

Create a category for assessable records

After you create a metric type and generate assessable records, create categories for the themes you are interested in using to evaluate the assessable records.

Role required: none

1. Navigate to Assesments > Metric Definition > Categories.
2. Click New to create a new metric category.
3. Complete the Metric Category form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the metric category. The name appears on assessment questionnaires.</td>
</tr>
<tr>
<td>Type</td>
<td>[Required] Metric type this category is used for. Metrics in this category can be used to evaluate assessable records for the type you select. This field becomes read-only after you save the record.</td>
</tr>
<tr>
<td>Create stakeholders</td>
<td>Check box that enables (selected) or disables (cleared) automatic stakeholder creation for this category. When the check box is selected, the system makes all associated category users stakeholders for each assessable record associated to this category. This saves the time required to manually create stakeholders. For more information, see Create Stakeholders Automatically. Existing stakeholders are not impacted if you clear the check box.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Table</td>
<td>[Read-Only] Table that contains the records to evaluate with metrics in this category. The system automatically populates this field with the same table as the metric type selected. The table does not appear in the field until you save the form. This field is visible depending on whether a <strong>Type</strong> is selected.</td>
</tr>
<tr>
<td>Weight</td>
<td>[Required] Numerical value that represents the importance of this category relative to other categories. A greater value indicates greater importance. For more information, see <a href="#">Weight categories and metrics</a>. The default value is 10.</td>
</tr>
</tbody>
</table>
| Scoring type | Scoring method for an attestation category. This field appears only when the **metric type** is an attestation with a **Scoring type** of **Percent**. Use this field to configure different scoring types for specific attestation categories. Possible selections are:  
  - **Percent**: Attestation score as a weighted percentage of correct answers for scored questions.  
  - **All or nothing**: All answers for attestation type questions must be correct or the score is zero for the entire category. |
<p>| Total metrics| [Read-Only] Number of metrics associated to this category. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Filter conditions that assessable records must meet to be evaluated using metrics in this category.</td>
</tr>
<tr>
<td></td>
<td>The filter operates on fields on the selected table.</td>
</tr>
<tr>
<td></td>
<td>If you specify a filter condition, the system automatically associates matching assessable records to the category when you save the record.</td>
</tr>
<tr>
<td></td>
<td>If you change the filter conditions, the system removes and creates assessable record associations as needed.</td>
</tr>
<tr>
<td></td>
<td>The system also deletes stakeholders for assessable records it disassociates.</td>
</tr>
<tr>
<td></td>
<td>The system does not remove assessable record associations created by users, even if the assessable records do not match the filter conditions.</td>
</tr>
<tr>
<td></td>
<td>This field is visible only when a <strong>Type</strong> is selected.</td>
</tr>
<tr>
<td>Description</td>
<td>Descriptive information about the category that appears on assessment questionnaires.</td>
</tr>
<tr>
<td>Details</td>
<td>Text field containing an HTML editor. Use this field to present important details about this category to assessment recipients. Details can include formatted lists, images, videos, or links to external websites. You may need to configure the form to see this field.</td>
</tr>
<tr>
<td>Related Lists</td>
<td></td>
</tr>
<tr>
<td>Assessment Metrics</td>
<td>All metrics associated to this category. There must be at least one metric associated to the category to use any assessments with the category.</td>
</tr>
<tr>
<td>Assessable records</td>
<td>All assessable records associated to this category. There must be at least one assessable record associated to the <strong>category</strong> to use any assessments with the category. This related list is hidden if the <strong>Evaluation method</strong> is <strong>Attestation</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Users</td>
<td>All users associated to this category. There must be at least one category user to create scheduled assessments using metrics in this category. This related list is available only when the associated metric type has schedule type set to <strong>Scheduled</strong>. Category users are not a part of the <strong>on-demand assessment</strong> process.</td>
</tr>
</tbody>
</table>

Assessable record associations

Only the assessable records associated to a category can be evaluated using metrics in that category. Manage which assessable records you evaluate for each category by creating and removing the associations.

Note: You can also manage which categories an assessable record is associated to using the **Assessable Record** form.

Manage an assessable record automatically

You must define at least one filter condition in the Metric Category form to automatically associate assessable records with the category.

Role required: assessment_admin or admin

1. Navigate to **Assessments > Metric Definition > Categories**.
2. Open a category.
3. In the **Filter** field, create a condition statement that defines the records you want to associate to this category. The available fields are from the table defined for the metric type.

   For example, for a category in the **Vendor** type, you might create the condition **[Country] [is] [Germany]**. Metrics in the category will be used only to evaluate vendors that meet this condition.
4. Save the record.

   The system associates this category to all assessable records that match the filter condition. As the system generates new assessable records that match the condition, it associates them to this category. If you change the filter condition, the system removes and creates assessable record associations, as appropriate. Note the following:
   - The system does not remove assessable record associations created manually, even if the assessable records do not match the filter conditions.
   - The system deletes any related stakeholders, if applicable, when it disassociates assessable records from a category.
   - The system does not create or remove assessable record associations if the related source records change. For example, consider the previous example in which all assessable records for vendors based in Germany were associated to a particular category. Assume that a user changes the **Country** of one of the vendor records from **Germany** to **Canada**. The system does not disassociate the related assessable record from the category, even though it no longer matches the filter condition.

   Note: To configure the system to automatically associate all assessable records from a metric type to a category, use the same category condition as the metric type condition. For example,
if the Vendor metric type has the condition [Vendor] [is] [true], create the same condition for the category. If the metric type does not have a condition specified, you can use the category condition [Sys ID] [is not empty], which includes all records from any table.

Manage an assessable record manually
You can manually create and remove assessable record associations for a category using a related list on the Metric Category form.

Role required: assessment_admin or admin
Keep in mind that when you disassociate an assessable record from a category, the system deletes any related stakeholders, if applicable, and you can no longer evaluate the assessable record using metrics in that category.

1. Navigate to Assessments > Metric Definition > Categories.
2. Open a category.
3. In the Assessable records related list, click Edit and use the slushbucket to add or remove one or more assessable records.

Only assessable records associated to this category’s metric type are available for selection.

Delete a category
When you delete a category, the system also deletes the associated category users and stakeholders.

Role required: assessment_admin or admin

1. Navigate to Assessments > Metric Definition > Categories.
2. Alternatively, navigate to a record that contains a related list for categories.
3. Open a record from the list.
4. On the Metric Category form, click Delete.
5. Click OK when asked to confirm the action.

The system displays messages at the top of the form describing the actions taken.

Assessment metrics
In the Assessments application, a metric is a trait or value used to evaluate assessable records.

Depending on the metric method, a metric can be used as either of the following:

- A question on assessment questionnaires, to obtain a subjective value such as how much people enjoy using each cell phone offered in the service catalog.
- A script that queries the database, to obtain an objective value such as the number of incidents related to each cell phone.

Each metric is associated to one metric category and can be used to evaluate assessable records for that category only. For example, the Support Rating category contains metrics that measure the quality of vendors’ customer support services. One metric addresses the quality of product documentation. Others address the strength of the user community, self-service options, online training, and other support services. Each metric has a numeric weight value to indicate its importance relative to other metrics in the same category.

Assessment administrators can create and administer metrics and metric templates, which define reusable sets of answer options for metrics used as assessment questions. If you decide to use metric templates, you may want to create them before you create metrics.

Note: Changes to assessment metrics and metric templates are reflected in assessment questionnaires and results, even if users have saved responses already. If possible, avoid
changing the original meaning of a metric question, the data type, or any metric templates in use by existing assessments.

Create a metric for a category

After you create a category, create metrics you can use to evaluate the assessable records for that category.

Role required: assessment_admin or admin

When you create metrics, consider the focus of the category and what characteristics you want to measure for the items you will assess. If you plan to use metrics for assessment questionnaires, you may want to review tips for writing good questions.

1. Navigate in one of the following ways:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessments &gt; Metric Definition &gt; Categories</td>
<td>Open a category, and locate the Assessment Metrics related list. This is the suggested navigation path, as the system links metrics to the category from which they are created.</td>
</tr>
<tr>
<td>Assessments &gt; Metric Definition &gt; Metrics</td>
<td></td>
</tr>
</tbody>
</table>

2. Click New.

3. Fill in the fields, as appropriate.

Table 80: Assessment Metric form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the metric.</td>
</tr>
</tbody>
</table>
| Category  | Metric category that the metric belongs to. The system automatically populates this category if you create a new metric from the Metric Category form.  

**Note:** You cannot change the category if the Depends on field is set or if another metric depends on this metric.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Determines how to use the metric.</td>
</tr>
<tr>
<td></td>
<td>• Assessment: Non-scripted metric. Make the metric available as a question on an assessment questionnaire. The Assessment method is compatible with all data types except <strong>Duration</strong>.</td>
</tr>
<tr>
<td></td>
<td>• Script: Scripted metric. Obtain values by writing a custom script. The Script method is compatible with the <strong>Duration</strong>, <strong>Number</strong>, and <strong>Percentage</strong> data types.</td>
</tr>
<tr>
<td></td>
<td>• Default answer from field: This option appears only if you have selected an <strong>Assessment Metric Type</strong> that contains a table. The <strong>General</strong> tab adds two fields:</td>
</tr>
<tr>
<td></td>
<td>• Default answer: Select the default answer for the question. The list comes from the selected table.</td>
</tr>
<tr>
<td></td>
<td>• Ask question: Specifies when to ask the question: always or only if the default answer is empty.</td>
</tr>
<tr>
<td></td>
<td>• Default answer from script: The <strong>General</strong> tab adds a field:</td>
</tr>
<tr>
<td></td>
<td>• Ask question: Specifies when to ask the question: always or only if the script does not contain a default answer. The script is defined on the <strong>Field Type</strong> tab.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you select a <strong>Data type</strong> that is incompatible with the selected <strong>Method</strong>, the system automatically changes the <strong>Method</strong> to the correct value.</td>
</tr>
<tr>
<td>Weight</td>
<td>Numeric value that represents the importance of this metric relative to other metrics in the same category. By default, the weight is 10. For weighting suggestions, see <strong>Weight Metrics and Metric Categories</strong>.</td>
</tr>
<tr>
<td></td>
<td>This field is visible and required unless the <strong>Data type</strong> is <strong>Date</strong>, <strong>Date/Time</strong>, or <strong>String</strong>. These data types are not included in results calculations.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Order</td>
<td>Numeric value that determines the order of the metric question on assessment questionnaires, relative to other metric questions in the same category. The metric with the smallest order value appears as the first question in the category's section. By default, the order is 100. For more information about questionnaire layout, see Complete Assessment Questionnaires. <strong>Note:</strong> It does not matter which order value you use for metrics with the Script method, as they do not appear on questionnaires.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that determines whether this metric is used to gather assessment results. If a non-scripted metric (Assessment method) is inactive, the question does not appear on assessment questionnaires generated after the metric becomes inactive. If a scripted metric (Script method) is inactive, the script does not run.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Check box that makes the metric question mandatory (selected) or optional (cleared) on assessment questionnaires. Users cannot submit questionnaires until they provide valid responses to all mandatory questions, which are denoted by a red field status indicator. This field is visible only if the Method is Assessment, the Depends on field is empty, and the data type is not Checkbox.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allow not applicable</td>
<td>Check box that determines whether to include a <strong>Not Applicable</strong> answer option for this metric question on assessment questionnaires. Users can select <strong>Not Applicable</strong> if they do not have sufficient information to respond to a question or if a question does not apply to a particular assessable record. User responses of <strong>Not Applicable</strong> are excluded from results calculations. This field is visible only if the Method is <strong>Assessment</strong> and certain data types are selected.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Additional fields you can add by customizing the form**

<table>
<thead>
<tr>
<th>Details</th>
<th>Text field containing an HTML editor. Use this field to present important details about this metric to assessment recipients. Details can include formatted lists, images, videos, or links to external websites.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**General Section**
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
<td>Text to use as the question on assessment questionnaires. Enter a clear, straightforward question that is easy to answer, such as <strong>How likely are you to recommend this vendor for the purchase of office supplies?</strong></td>
</tr>
<tr>
<td>Description</td>
<td>Information about the metric and what it evaluates. If the Method is <strong>Assessment</strong>, include details that help users understand how to answer the question, as this text appears as a hint when a user points to the question text on the questionnaire.</td>
</tr>
<tr>
<td>Depends on</td>
<td>Setting used to make this metric a conditional question. Select an existing metric question from the list, which displays <strong>Checkbox, Choice, Likert Scale, Template</strong> and <strong>Yes/No</strong> metrics of the same category as this metric. Then, use the <strong>Displayed when</strong> field to set the conditions that display this metric question on questionnaires. The system prevents the creation of recursive dependencies between metrics. For example, if Metric A depends on Metric B, Metric B cannot depend on Metric A. This field is visible only if the Method is <strong>Assessment</strong>.</td>
</tr>
<tr>
<td>Displayed when</td>
<td>Answer options for the selected <strong>Depends on</strong> metric question that, when chosen on questionnaires, display this metric question. This field is visible and required only if the <strong>Depends on</strong> field is set.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data type</td>
<td>[Required] Format of the expected response data. The function of the data type depends on the selected Method. If the method is Assessment, the data type determines how users answer the corresponding question on questionnaires. If the method is Script, the data type determines how the system calculates assessment results. Note: You cannot change the data type if another metric depends on this metric.</td>
</tr>
<tr>
<td>Randomize answers</td>
<td>Check box that determines whether to present the answer options for this metric question in a random order each time a user opens an assessment questionnaire that contains the question. Answer preference is sometimes influenced by the order in which answer options appear, which can result in biased results. Randomizing answer options can help prevent this bias. This field is visible only if a data type that requires metric definitions is selected. Note: Randomizing answer options for certain questions may make those questions confusing for the person answering. In general, only randomize answer options that do not follow a logical order. For example, the following question is confusing when randomization is enabled:</td>
</tr>
<tr>
<td>Template</td>
<td>Metric template to use for the metric question. A metric template is a set of predefined answer options. This field is visible and required only if the Data type is Template. Note: You cannot change the template if another metric depends on this metric.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dependent plugin</td>
<td>Plugin that contains the tables queried in the script. The system executes the metric script only if the plugin is active. The default available values are <strong>Asset Management</strong>, <strong>CMDB</strong>, <strong>Core</strong>, <strong>Cost Management</strong>, <strong>Procurement</strong>, and <strong>Software Asset Management</strong>. You must select a plugin if the Method is <strong>Script</strong>. This field is visible only if the Method is <strong>Script</strong>.</td>
</tr>
<tr>
<td>Note:</td>
<td>An administrator may need to add more <em>choices</em> of plugins to the field.</td>
</tr>
<tr>
<td>Scale definition</td>
<td>Setting that determines whether lesser or greater numerical values equate to a good score in assessment result calculations. Select <strong>Low</strong> if lesser numerical values are better, such as for a metric that measures the number of incidents for a vendor. Select <strong>High</strong> if greater numerical values are better, such as for a metric that measures user satisfaction on a scale of one to five. The default value is <strong>High</strong>. This field is visible and required unless the Data type is <strong>Date</strong>, <strong>Date/Time</strong>, or <strong>String</strong>. The results for these data types are not included in results calculations. Note: For information about how to set the scale definition for data types that do not require you to set a numerical value, see <strong>Data Types</strong>.</td>
</tr>
<tr>
<td>Min</td>
<td>Lowest numerical value to be used as an answer option on assessments or as a scaled value in a scripted metric. This field is visible and required only if certain data types are selected. If the data type is <strong>Choice</strong> or <strong>Likert Scale</strong>, this field is read-only and is set automatically based on the smallest metric definition Value.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>Highest numerical value to be used as an answer option or scaled value. This field is visible and required only if certain data types are selected. If the data type is <strong>Choice</strong> or <strong>Likert Scale</strong>, this field is read-only and is set automatically based on the largest metric definition Value.</td>
</tr>
</tbody>
</table>

| Script    | Script that obtains the desired system information. This field is visible and required only if the Method is **Script**.                                                                                                                                                                           |

| Related List | Lists all metric definitions, which appear as answer options for questions on assessment questionnaires. Click **New** to create a new metric definition. This related list is available only if the Data type is **Choice** or **Likert Scale**. |

4. **Save the record.**

When you create a **Choice** or **Likert Scale** question, you must reopen the Assessment Metric form after you submit it to *Create a metric definition*. If you distribute a questionnaire without creating the answers for questions with these data types, recipients are unable to answer the questions. If the questions are mandatory, the recipients are unable to submit their questionnaires.

**Create a metric definition**

Metrics with Choice or Likert Scale data types must have metric definitions defined to represent answer options on questionnaires.

**Role required:** assessment_admin or admin

Each metric definition appears as one answer option for a question on an assessment questionnaire. For example, a metric for the question "How satisfied are you with the quality of this vendor?" has three metric definitions to create these answer options: **Very satisfied**, **Somewhat satisfied**, and **Not at all satisfied**.

1. Open the metric record you want to edit.
2. In the **Assessment Metric Definitions** related list, click **New** to create a new metric definition.
3. Fill in the fields:
   - **Display**: Enter the text you want to appear as the answer option.
   - **Value**: Enter the numeric value, greater than or equal to zero, to which the answer option equates. Values are used to calculate metric results. When you view assessment questions, metric definitions appear in order from least to greatest value.
     - The system sets the minimum and maximum values according to the metric definition values.
     - To prevent reporting confusion, no two metric definitions for a given metric should have the same Value.
4. Click **Submit**.
Change the order of an answer

By default, the order of answer options is established when a question is created with a data type of **Choice** or **Likert Scale**.

Role required: assessment_admin or admin

The system creates an Order value for each choice based on where it appears in the original list when you submit it. This order corresponds to the Value setting you give the answer. The choice with the lowest order number appears first, and the choice with the highest order number appears last. If you want to reconfigure the order of the answer options, you must do so by editing the Order field and not the Value field.

The Order column is hidden in the base system.

To edit these records, configure the Assessment Metric Definitions related list in the question record to show the Order column, and then edit the order number in the list. The Value setting remain the same for each answer option after reorganization.

Metric types and assessable records

In the Assessments application, assessment administrators create and administer metric types and assessable records.

A metric type defines a set of records an organization wants to evaluate, such as vendors, projects, or employees. For each type, the system generates unique assessable records that link the type to records that need to be evaluated, such as the individual records for the vendors Amazon and Intel. There may be multiple assessable records for the same source record if the source record meets the criteria for more than one type. For example, you might want to evaluate a record on the Company table, such as Intel, as a vendor and as a manufacturer, with different categories and metrics.

For configuration suggestions, see Administrator tasks on page 312.

Create metric types and generate assessable records

Each metric type sets a table and filter conditions that define a set of records to evaluate.

Role required: assessment_admin or admin

For example, the Vendor Performance application provides the Vendor metric type, which defines an assessable record for each source record in the Company table that has the Vendor field selected. Create a metric type for each set of records you want to evaluate, such as vendors, users within an organization, or projects.

In addition to defining assessable records, metric types also determine how to filter decision matrix data and set the schedule type, which determines whether assessments are on-demand or scheduled.

1. Navigate to Assessments > Metric Definition > Types.
2. Click New.
3. Fill in the fields, as appropriate (see table).
4. When you are satisfied with the type settings, click **Generate Assessable Records** to save the record and create assessable records. The page redirects to the list of assessable records created.

You must click **Generate Assessable Records** to create the initial set of assessable records based on the table and conditions specified on the type record. After that, the system creates an assessable record each time a new matching record is created. **Generate Assessable Records** is located under Related Links.

Note: Clicking **Generate Assessable Records** can trigger the deletion of existing assessable records under certain circumstances. For more information, see *Enforce a condition to delete an assessable record* on page 350.

### Table 81: Metric types

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Required] Name of the type. It may be practical to give the type a name that indicates which records it defines.</td>
</tr>
<tr>
<td>Assessment duration</td>
<td>Amount of time assessors have to complete their assigned questionnaires, starting from the time the assessment is generated. The assessment duration sets the <strong>Due date</strong> that appears on assessment instances. Changes made to the duration of an attestation in the GRC Control Test Definition form are updated dynamically in this field. The default duration is 14 days.</td>
</tr>
<tr>
<td>Live feed</td>
<td>Check box that determines whether to enable (selected) or disable (cleared) the Live Feed view in scorecards for all assessable records created from this type.</td>
</tr>
<tr>
<td>Notify manager if overdue</td>
<td>Check box that determines whether to send (selected) email notifications a user’s manager when that user fails to submit an assigned questionnaire on time.</td>
</tr>
<tr>
<td>Assessment manager</td>
<td>Manager for this assessment metric type. These users are responsible only for managing the assessment process and not the results. The system notifies the assessment manager when an assessment for this metric type is past due.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Scoring type</td>
<td>Scoring method to use for questions with a data type of attestation. The possible selections are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Percent</strong>: Attestation score as a weighted percentage of correct answers for scored questions. When you select this scoring type, categories can be scored as <strong>Percent</strong> or <strong>All or nothing</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>All or nothing</strong>: Requires all answers for attestation type questions to be correct, or the score is zero for the entire attestation. When you select this scoring type, all categories are scored automatically as <strong>All or nothing</strong>.</td>
</tr>
<tr>
<td>Scale factor</td>
<td>[Required] Number to represent the best possible score for assessment results. All results for assessments of this type are scaled to this number. 10 is generally a good scale factor.</td>
</tr>
</tbody>
</table>

**Note**: This field becomes read-only when it contains a value and you save the metric type. Choose a scale factor you are satisfied with before you save the metric type.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles</td>
<td>Additional user roles that can view the results and access records associated with this type. Users with the specified roles have read access to this type record as well as to associated categories, metrics, assessable records and scorecards, category users, stakeholders, and decision matrices. <strong>Note:</strong> Users with these roles do not have access to Assessments modules unless they are also assessment administrators. Users with these roles can navigate to the records by other means, such as from reference fields on assessment instances. This field provides the option to easily grant certain users access to specific assessment data in special cases. For example, the <strong>Vendor</strong> metric type provides access to users with the vendor_manager role so they can view results and compare assessable records when they open scorecards or decision matrices in the Vendor Performance application.</td>
</tr>
<tr>
<td>Signature</td>
<td>[Optional] Acknowledgement by a survey recipient of requirements, admonitions, or expectations related to an assessment. For more information, see Create an assessment signature on page 347.</td>
</tr>
<tr>
<td>Return URL</td>
<td>Destination address of a web page that is presented to users after they submit an assessment for this metric type. When a return URL is configured, the end note content does not appear.</td>
</tr>
<tr>
<td>State</td>
<td>[Read-Only] Status of the assessment: Draft or Published.</td>
</tr>
<tr>
<td>Business rule</td>
<td>[Admin only] Business rule the system creates to monitor the specified table. When a new record is added to the table that meets the metric type conditions, the business rule generates a corresponding assessable record. If someone changes the metric type’s table or conditions, the system updates the business rule to reflect the changes. Existing assessable records associated with the metric type are not affected.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Schedule type</td>
<td>Setting that determines which assessment process to use. Select <strong>On demand</strong> or <strong>Scheduled</strong>.</td>
</tr>
<tr>
<td>Scheduled job</td>
<td>[Admin only] Scheduled job the system creates to generate assessments for this metric type. This field is populated the first time you save the record. The scheduled job name follows the format <code>&lt;type name&gt;-Assessment Creation</code>. Administrators can configure a recurring assessment generation schedule for the metric type. This field is visible only when the schedule type is <strong>Scheduled</strong>. <strong>Note:</strong> If you change the schedule type from <strong>Scheduled</strong> to <strong>On demand</strong>, the system deletes the scheduled job. If you change the schedule type back to <strong>Scheduled</strong>, the system creates a new scheduled job. The previous assessment generation schedule is not saved, but can be reconfigured if desired.</td>
</tr>
</tbody>
</table>
| Evaluation method  | [Admin only] Indicates the evaluation method for this metric type. The system sets this field based on how the metric type is created. The possible evaluation methods are:  
  • **Assessment**: Default filter value for metric types. Only this evaluation method uses **assessable records**, conditions, and **stakeholders**.  
  • **Attestation**: For details, see **Attestations**.  
  • **Survey**: For details, see **Survey definitions** on page 576.                                                                                                                                                                                                                                                                                                                   |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale factor</td>
<td>[Required] Number to represent the best possible score for assessment results. All results for assessments of this type are scaled to this number. <strong>10</strong> is generally a good scale factor.</td>
</tr>
<tr>
<td>Note:</td>
<td>This field becomes read-only when it contains a value and you save the metric type. Choose a scale factor that you are satisfied with before you save the metric type.</td>
</tr>
<tr>
<td>Allow retake</td>
<td>Check box that allows users to modify their answers to a completed assessment, quiz, survey, or attestation. After the due date, the system removes the assessment from the user's My Assessments and Surveys page.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that determines whether the metric type is active. When the metric type is inactive (cleared), assessment generation is disabled and users cannot take existing assessments.</td>
</tr>
<tr>
<td>Description</td>
<td>Helpful information about this type. Enter a clear description of the type and its purpose.</td>
</tr>
<tr>
<td>Introduction</td>
<td>Introductory content to display on assessment questionnaires. Consider adding a company logo, a welcome message, background information about the survey, or instructions. You may need to configure the form to see this field.</td>
</tr>
<tr>
<td>End note</td>
<td>Content to display on the screen that appears when someone submits an assessment questionnaire. Consider adding a thank you message, followup instructions, or other applicable information. You may need to configure the form to see this field. End notes are not displayed if a <strong>Return URL</strong> is specified.</td>
</tr>
<tr>
<td>Send notifications</td>
<td>Select the check box to send notifications for this assessment, survey, or attestation. You may need to configure the form to see this field.</td>
</tr>
<tr>
<td>Condition section</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Table</td>
<td>[Required] Table that contains the records you want to evaluate. The system creates assessable records for records on this table that meet the conditions you specify, if any. The number of matching records appears as a link by the <strong>Condition</strong> field. The link dynamically updates if you change the table selection. Click the link to open the list of matching records in a new tab or window.</td>
</tr>
<tr>
<td>Enforce condition</td>
<td>Check box that determines what happens to assessable records when you change the selected table or conditions.</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition builder that defines specific records to assess from the selected table. If you do not specify any conditions, the system creates assessable records for all records on the selected table. Click the refresh icon to update the adjacent record count.</td>
</tr>
<tr>
<td>Decision Matrix section</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Additional roles are required to view the records on certain tables. If you select a table that you do not have access to, a warning message appears by the **Condition** field where the number of matching records would be. You cannot generate assessable records for tables you do not have sufficient roles for.

**Note:** If you change the table or conditions, you must click **Generate Assessable Records** to create new assessable records.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter field</td>
<td>[Required] Field on the selected table that can be used to filter results on decision matrices and scorecards. For example, if the selected table is <strong>Company [core_company]</strong> and you choose <strong>Vendor type</strong> as the filter field:</td>
</tr>
<tr>
<td></td>
<td>• When you view decision matrices for this type, the decision matrix menu to filter plotted items lists vendor types: <strong>Applications</strong>, <strong>Hardware</strong>, <strong>Services</strong>, and <strong>Software</strong>.</td>
</tr>
<tr>
<td></td>
<td>• On scorecards for this type, the Averages view displays ratings by vendor type.</td>
</tr>
<tr>
<td>Filter condition</td>
<td>Condition builder that defines which records from the table are available as choices on the filter menu. For example, if the filter field is <strong>Vendor type</strong> and you create the filter condition <strong>[Name] [does not contain]</strong> <strong>[Hardware]</strong>, only records with vendor types of <strong>Applications</strong>, <strong>Services</strong>, and <strong>Software</strong> are possible choices as decision matrix filters.</td>
</tr>
</tbody>
</table>
Geneva ServiceNow ServiceNow Platform

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default matrix filter</td>
<td>Record to use as the default filter choice on decision matrices and scorecards. The selected Filter field and Filter condition control the available record choices. To see the records, if any, click the refresh icon next to the field. Consider the case that the filter field is Vendor type and you select Applications as the default matrix filter. The filter choice on decision matrices for this type is set to Applications by default. If you change the value of the Table or Filter condition field, you must click the refresh icon to view the updated Default matrix filter choices. If you do not, the system selects the first available choice from the updated field choices when you save the record.</td>
</tr>
</tbody>
</table>

Related Lists

<table>
<thead>
<tr>
<th>Metric Categories</th>
<th>All metric categories associated with the type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessable Records</td>
<td>All assessable records associated with the type. This related list is hidden in the Attestation view of this form.</td>
</tr>
</tbody>
</table>

**Note:** The Generate Assessments button is visible to administrators in certain cases for scheduled assessments. Do not click the button until after completing the prerequisite tasks.

Delete a metric type

Deleting a metric type entails deleting many related records.

Role required: assessment_admin or admin

You must delete some of these records manually before deleting the type, while the system deletes others automatically with the type.

1. Delete the records associated with the type to delete:
   - Assessment results (metric and category results)
   - Assessment instance (questions and assessment instances, in that order)
   - Assessment groups

2. Delete the type.
   A confirmation dialog box appears and alerts you that certain records associated with the type will also be deleted.

3. Click OK to delete the type and these related records:
   - Scheduled job for assessment generation
   - Business rule for assessable record generation
   - Assessable records
   - Metric categories
   - Category users
Export an assessment
You can share assessments between ServiceNow instances by exporting an assessment and then importing the assessment on another instance.
Role required: assessment_admin or admin

Note: The system exports a single XML file that does not contain result data.

The XML file contains a metric type [asmt_metric_type] and the following records that are associated with the type:
• Assessable records [asmt_assessable_record]
• Metric categories [asmt_metric_category]
• Metrics [asmt_metric]
• Metric definitions [asmt_metric_definition]
• Category users [asmt_m2m_category_user]
• Stakeholders [asmt_m2m_stakeholder]
• Decision matrixes [asmt_decision_matrix], [asmt_m2m_xcategory_matrix], and [asmt_m2m_ycategory_matrix]

1. Navigate to Assessments > Metric Definition > Types.
2. Right-click the record and select Export Assessment.
3. Save the XML file.

Import an assessment
Share assessments between ServiceNow instances by importing a previously exported assessment.
Role required: admin
1. Ensure that the target instance has assessments enabled.
2. Follow the procedure detailed in Import a Record as XML Data.

Create an assessment signature
A signature on an assessment questionnaire contains assertions that can communicate directions, a legal statement, or any text that you want the recipient to consider.
Role required: assessment_admin or admin

You can require the recipient to select a check box or provide a full signature to acknowledge acceptance of the assertion before submitting the form. You can display assertions without requiring a signature. An assessment property called Require authentication for user signature allows you to require users to provide their user name and password when an assessment asks for a full name signature.

1. Navigate to Assessments > Metric Definition > Signatures.
   All signatures in the system appear in this list.
2. Click New.
3. Complete and submit the form using the fields in the table.
### Table 82: Assessment signature fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Descriptive name for this signature</td>
</tr>
<tr>
<td>Signature type</td>
<td>Type of signature required. The selections are <strong>Checkbox</strong>, <strong>Full name</strong>, or <strong>Assertion only</strong>. If <strong>Assertion only</strong> is selected, no signature is required to submit the questionnaire. By default, the system requires user authentication for a full name entry.</td>
</tr>
<tr>
<td>Assertion</td>
<td>Text to present to recipients that requires acknowledgement.</td>
</tr>
</tbody>
</table>

Assessable records

An assessable record links a source record you want to evaluate, such as the company record for Amazon or the user record for a sales representative, to a metric type, such as vendors or employees.

You use assessments to evaluate the assessable record. The system generates assessable records from the source records that match the table and conditions set on the Assessment Metric Type form. You evaluate the assessable records with metric categories and metrics, which define traits and values to assess. For metric types with the **On demand** schedule type, you can [generate on-demand assessments](#) from the Assessable Record form. This method of assessment generation makes it easy to create and preview short questionnaires or to quickly obtain assessment results for specific assessable records.

View an assessable record

View the Assessable Record form to edit preferences and perform various actions.

Role required: assessment_admin or admin

1. Navigate to **Assessments > Assessable Records**.
2. Open a record from the list.
   By default, the list displays only assessable records with **Active** metric types.
3. On the **Assessable Record** form, edit fields and perform other actions as necessary (see table).

### Table 83: Viewing Assessable Records

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Read-Only] Assessable record name based on the <strong>display value</strong> of the source record. The name appears on assessment questionnaires.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Source</td>
<td>[Read-Only] Source record the assessable record is linked to. The <strong>Source</strong> reflects the table name and source record display value. For example, if the <strong>Name</strong> field is the display value for the Company table, the assessable record for a company record named Amazon has the <strong>Source</strong> value <strong>Company: Amazon</strong>.</td>
</tr>
<tr>
<td>Live feed</td>
<td>Check box that, when selected, creates a live feed group for the assessable record, which appears on the scorecard. If you clear the check box after a live feed group has been created, the system deletes the live feed group and all its messages.</td>
</tr>
<tr>
<td>Type</td>
<td>[Read-Only] Metric type from which the assessable record was generated.</td>
</tr>
<tr>
<td>Decision matrix</td>
<td>Check box that, when selected, enables this assessable record's results data to appear on decision matrixes of the same metric type. Decision matrixes are graphs that plot the assessment results for multiple assessable records. If you clear the check box, the assessable record still appears on the default decision matrixes if you click the <strong>View Matrix</strong> related link on the assessable record's scorecard.</td>
</tr>
<tr>
<td>Live feed group</td>
<td>[Read-Only] Live feed group for this assessable record. When you select the <strong>Live feed</strong> check box and save the record, the system populates this field.</td>
</tr>
<tr>
<td>Related Links</td>
<td></td>
</tr>
<tr>
<td>View Scorecard</td>
<td>Opens the scorecard for the assessable record.</td>
</tr>
<tr>
<td>Related Lists</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>All metric categories associated with the assessable record. An assessable record must be associated to a category to be evaluated. Click <strong>Edit</strong> to add or remove category associations. Note that it is often more efficient to associate assessable records to categories using the Metric Category form.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category users</td>
<td>All stakeholders who can take assessment questionnaires about this assessable record. Click <strong>Edit</strong> to create and delete stakeholders. For more information about this and other methods of creating stakeholders, see <a href="#">Stakeholders</a>. This related list is available only when the associated metric type has the <strong>Scheduled</strong> schedule type.</td>
</tr>
</tbody>
</table>

**Enforce a condition to delete an assessable record**

By default, the system does not delete assessable records, even if you change the table or conditions for the type and the existing assessable records no longer match.

Role required: assessment_admin or admin

The system can be configured so you can trigger the deletion of assessable records that do not match the type table and conditions.

1. On the Assessment Metric Type form, select the **Enforce condition** check box to enable deletion of existing assessable records of this type that do not match the table or conditions specified.

   When the check box is cleared, the system retains existing assessable records of this type, even if they do not match the table and conditions specified.

2. Click **Generate Assessable Records** to delete all assessable records associated with the type that do not match the current table and conditions.

   The system does not automatically delete assessable records, even if the **Enforce condition** check box is selected. You must click **Generate Assessable Records** with the check box selected every time you want the system to delete non-matching assessable records.

**Delete an assessable record**

When you delete an assessable record, the system deletes any stakeholders for the record.

Role required: assessment_admin or admin

**Note:** If a source record is deleted, the system deletes the associated assessable record. To delete the source record, you must first delete all associated metric results and category results.

Delete the assessable record:

- To delete a single record, open the record and click **Delete**.
- To delete multiple records, use the Assessable Records list.

**Create a metric template**

Create a metric template to define reusable rating scales for evaluating non-scripted metrics.

Role required: assessment_admin or admin

1. Navigate to **Assessments > Metric Definition > Templates**.
2. Click **New**.
3. Fill in the **Name** field on the **Assessment Metric Template** form and save the record.
4. Under the **Assessment Template Definitions** related list, click **New**.
   
   Create a template definition for each answer option you want to appear on an assessment question.
5. Fill in the following fields:
   - **Display**: Enter the text to appear as the answer option.
   - **Value**: Enter a numeric value, equal to or greater than zero, that you want the answer option to equate to. Values are used in results calculations.

   When you view assessment questions that use templates, answer options appear in order from least to greatest Value.

   **Tip**: To prevent reporting confusion, no two template definitions for a given template should have the same Value.

6. Click **Update**.

**Metric templates**

Metric templates define reusable rating scales for evaluating non-scripted metrics. Each answer option on the scale is defined as a template definition, much like a metric definition. For example, the metric template named Satisfaction contains the template definitions: **Very Satisfied**, **Satisfied**, **Neutral**, **Dissatisfied**, and **Very Dissatisfied**.

Metric templates are available for metrics that have Template as the Data type. The following metric templates are available by default.

<table>
<thead>
<tr>
<th>Name</th>
<th>Assessment template definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likert 5</td>
<td>Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>Very Dissatisfied, Dissatisfied, Neutral, Satisfied, Very Satisfied</td>
</tr>
<tr>
<td>Frequency</td>
<td>Never, Seldom, Sometimes, Most of the time, All of the time</td>
</tr>
<tr>
<td>Amount</td>
<td>None, Few or little, Average amount, Many, Quite a lot</td>
</tr>
<tr>
<td>Size</td>
<td>Very Small, Small, Average, Large, Very Large</td>
</tr>
<tr>
<td>Quality</td>
<td>Very Poor, Poor, Average, Good, Very Good</td>
</tr>
<tr>
<td>Complexity</td>
<td>Very Complex, Complex, Moderate, Simple, Very Simple</td>
</tr>
</tbody>
</table>

**Update a metric minimum and maximum value to match a template**

For assessment results to be properly calculated, the Min and Max values of a metric that uses a template must be equal to the smallest and largest template definition Value, respectively.

**Role required**: assessment_admin or admin

The system sets the **Min** and **Max** fields automatically when you create a metric of the Template data type, based on the template definition values. However, the system does not update these fields for existing metrics if you add a new template definition to a template or if you update the Value of an existing template definition. If the new Value is less than the minimum value of any metrics that use the template or greater than the maximum value, update the metrics accordingly.
To edit the minimum and maximum values for a question that has the Template data type:

1. Navigate to Assessments > Metric Definition > Metrics.
2. Configure the list to show the Min and Max columns.
3. Run the following list filter condition: [Template] [is] [<select the template you updated>]
4. Ensure the Min and Max values match the smallest and largest template definition Value for the selected template, respectively.
   
   If the values do not match, edit the Min and Max directly from the list.

Note: When the data type is Template, a UI policy prevents the editing of Min and Max from the form.

Category users and stakeholders

Users who take assessments are individuals in your organization who have specific knowledge of the assessment categories and the records being assessed. A person who is qualified to answer assessment questions from metrics in a specific category becomes a category user when associated to that category. A category user then becomes a stakeholder when associated to a specific assessable record.

The system determines which assessable records and questions a user receives by looking at the metric categories and assessable records for which those users are stakeholders. Assessment administrators can create and manage category users and stakeholders.

Note: Category users and stakeholders are used only for scheduled assessments.

Category users

Category users are users who can potentially answer assessment questions about a particular metric category for scheduled assessments. Category users should have special knowledge of the categories and the metrics those categories contain. A category user becomes a stakeholder when associated to an assessable record. Create at least one category user for each category you want to use for assessment questionnaires. There is no need for category users if the category contains only scripted metrics.

Stakeholders

A stakeholder is a category user with specific product or service knowledge, who is associated to an assessable record. A stakeholder is familiar with the assessable record and all the metrics within a specific category. The same category user can be associated to multiple assessable records, in which case the system creates a stakeholder record for each association.

There are multiple ways to create stakeholders:

- Configure the system to create stakeholders automatically.
- Associate multiple category users to multiple assessable records.
- Associate multiple category users to one assessable record.
- Associate one category user to one assessable record.

Note: Ensure you establish the desired user and assessable record associations to categories before attempting to creating stakeholders.

Create a category user

Create category users by associating users in the system to metric categories.

Role required: assessment_admin or admin
When you create category users, choose users who are knowledgeable and can answer questions from that category. For example, a purchasing agent is a good category user for metrics that evaluate a vendor's discount practices.

1. Navigate to **Assessments > Metric Definitions > Categories**.
2. Open a category record.

   **Note:** The category must be associated to a metric type that has the **Schedule type** set to **Scheduled**.

3. In the **Users** related list, click **Edit**.
4. Select the appropriate users for this category.

Create stakeholders automatically
You can configure the system to create stakeholders automatically from all category users and assessable records associated with a category using the **Create stakeholders** check box on the Metric Category form.

Role required: assessment_admin or admin

If you associate a user to the category, the system makes the resulting category user a stakeholder for all the assessable records associated to the category. Similarly, if you associate a new assessable record to the category, the system makes all category users stakeholders for that record if they are not already stakeholders.

   **Note:** Use the **Create stakeholders** check box with caution. You can end up with more stakeholders than you want and a large amount of unnecessary assessment data. Automatic stakeholder creation is best suited to assessment implementations in which category users are knowledgeable about all or most of the assessable records associated to each category.

1. Navigate to **Assessments > Metric Definition > Categories**.
2. Select a category.
3. Associate category users and assessable records to the category.
4. Select the **Create stakeholders** check box.
5. Click **Update**.
   All the category users are associated to all the assessable records as stakeholders.
6. In the **Assessable Records** related list, click **Edit**.
7. Using the slushbucket, associate an assessable record to the category.

8. Click **Save**.
The system makes all category users stakeholders for the new assessable record.

9. Delete stakeholders as needed.

Associate multiple category users to multiple assessable records
The stakeholder list helper in the create stakeholders module is the most efficient way to associate multiple category users to multiple assessable records in a single interface.
Role required: assessment_admin or admin
You can select category users from one category at a time.

Note: You cannot edit or delete stakeholders using the list helper.

1. Navigate to Assessments > Advanced > Create Stakeholders.
The stakeholder list helper appears.
2. Select a metric type from the list of available types.
   
   **Note:** Only metric types for scheduled assessments are available. On-demand assessments do not use category users or stakeholders.

3. Select a category from the list of available categories.
   
   **Note:** Only categories within the selected metric type are available.

   The system populates the **Category User** and **Assessable Records** lists with category users and assessable records associated to the selected category.

4. Select one or more category users from the **Category Users** list.

5. Select one or more assessable records from the **Assessable Records** list.

6. Click the **Associate** arrow between the lists to complete the association.
   A message above the list helper advises you that the selected category users are now stakeholders for the selected assessable records.

Delete a stakeholder for multiple assessable records
You can delete stakeholders for multiple assessable records.

Role required: assessment_admin or admin

1. Navigate to **Assessments > Advanced > Assessment Stakeholders**.

2. Sort the list by **Assessable Record** or **Category User** depending on the perspective you want.

3. Select the check box for each stakeholder you want to delete.

4. Select **Delete** in the **Actions** choice list.

5. Click **OK** to confirm the action.
   The system deletes the stakeholders.
Associate one category user to one assessable record
You can create a single stakeholder using the list in the Assessment Stakeholders module.

Role required: assessment_admin or admin

Associate any category user to any assessable record.

1. Navigate to **Assessments > Advanced > Assessment Stakeholders.**
2. Click **New.** On the Assessment Stakeholders form, fill in these fields:
   - **Category user:** Select a category user.
   - **Assessable record:** Select an assessable record.
3. Click **Submit.**
   The selected category user becomes a stakeholder for the selected assessable record. To verify the association, navigate to the assessable record and note that the selected category user is on the Category users related list.

   ![Assessment Stakeholders](image)

   **Note:** Do not create stakeholders from category users and assessable records of different metric types.

Associate multiple category users to one assessable record
You can create many stakeholders for a single assessable record from the Assessable Record form.

Role required: assessment_admin or admin

You can also edit or delete stakeholders from the form.

1. Navigate to **Assessments > Assessable Records.**
2. Open a record from the list.
3. In the **Category users** related list, click **Edit.**
   These category users are the stakeholders for this assessable record.
4. Use the slushbucket to edit the category users associated to this assessable record. Add category users to create stakeholders. Remove category users to delete stakeholders.

The system only shows category users for categories associated to the assessable record. For example, if an assessable record is only associated to the Compliance Score and Support Rating categories, you do not see User Satisfaction category users.

Delete a stakeholder for one assessable record
You can delete stakeholders for one assessable record.

Role required: assessment_admin or admin

1. Navigate to Assessments > Assessable Records.
2. Open an assessable record.
3. In the **Category Users** related list, select the check box for each stakeholder you want to delete.
4. Select **Delete** in the **Actions** choice list.
5. Click **OK** to confirm the action.
The system deletes the stakeholders.

*Generate an on-demand assessment*
Use on-demand assessments to familiarize yourself with the basic assessment process and test your questionnaires using minimal configuration.

Role required: assessment_admin or admin

Select the **assessable records** to evaluate, create the categories and questions, and then assign an assessment to a user in the system. Pre-configured **stakeholders** are not used for on-demand assessments.

1. Create a **metric type** and set the **Schedule type** to **On demand** to allow for testing of your assessment configuration.
2. Generate the **assessable records** for the metric type you created.
3. Create the **metric categories** required to evaluate the assessable records selected.
4. Create one or more assessment questions, or **metrics**, for each category.
5. **Publish the assessment.**
6. Perform an **on-demand assessment** to test your categories and metrics.
7. Analyze the assessment ratings in an **assessment scorecard** or **decision matrix**.

On-demand assessments

On-demand assessments can be generated for metric types with the Schedule type field set to **On demand**.

For the system to properly generate on-demand assessments, the metric type must be active and published. The metric type must also be associated to at least one metric category. That metric category must be associated to one or more of each of the following items:

- Assessable record
- Metric

By default, an assessment administrator can generate an on-demand assessment for one assessable record or **multiple assessable records**.

Generate an on-demand assessment for one assessable record

When you generate an on-demand assessment from the Assessable Record form, the resulting assessment contains questions from the categories associated to the assessable record.

Role required: assessment_admin or admin

1. Navigate to **Assessments > Assessable Records**.
2. Open an assessable record associated to a metric type that has the **On demand** schedule type.
3. On the Assessable Record form, click **Assign Assessment**.
The Select Assessor dialog box appears.

4. Select a user from the Recent Assessors list or select a different user.
5. Click OK.

The system generates an assessment instance assigned to the selected user.

Generate an on-demand assessment for multiple assessable records

When you generate an on-demand assessment from the Assessment Metric Type form, the resulting assessment contains questions from all categories associated to any assessable records for the metric type.

Role required: assessment_admin or admin

For example, if there are 15 assessable records associated to at least one category with assessment metrics, the assessment contains questions for all 15 assessable records.

1. Navigate to Assessments > Metric Definition > Types.
2. Open a metric type that has the On demand schedule type.
3. On the Assessment Metric Type form, click Assign Assessment.

The Select Assessor dialog box appears.

4. Select a user from the Recent Assessors list or select a different user.
5. Click OK.

The system generates an assessment instance assigned to the selected user.

Generate an assessment with the on-demand API

The Assign Assessment buttons call an API to generate on-demand assessments.

Role required: assessment_admin or admin

1. To generate an assessment, call the AssessmentCreation API from any script in the system, such as a business rule or client script.
2. Use the sys_ids of these items:
   - An assessable record to evaluate. The assessable record must be associated with an on-demand type and at least one category that contains non-scripted metrics.
   - One or more users who should perform the assessment. The API generates a questionnaire about the specified assessable record for each user.
Schedule an assessment
After you have evaluated your questionnaires using on-demand assessments, edit your categories and metrics as needed, reset your metric type record, and select the users who are qualified to evaluate the assessable records.

Role required: assessment_admin or admin
1. Open the metric type you created for the on-demand assessment and set the Schedule type to Scheduled.
2. Make sure the categories and metrics you created for the on-demand assessment are correct.
3. Create category users who have special knowledge of your categories.
4. Create stakeholders by associating category users to specific assessable records.
5. Set a schedule for automatic assessment generation or generate the assessment manually. This procedure must be done by a system administrator.
6. Configure email notifications to remind users of their assigned assessments and to report to managers when an employee misses an assessment deadline.
7. Analyze the assessment ratings in an assessment scorecard or decision matrix.

Schedule types
You can schedule assessments for preconfigured users or send them to any user on demand.

On-demand assessments require less setup. Scheduled assessments define users, or stakeholders, who have specific knowledge of the records being evaluated, and automatically send out assessment questionnaires to those users at scheduled intervals. The best practice is to configure an on-demand assessment first to familiarize yourself with the process and test your questionnaires. When you are satisfied with your categories and questions, reconfigure for a scheduled assessment. These configuration paths are the easiest for both assessment types. Refer to Key Terms for definitions of the terms used in these procedures.

Set an assessment generation schedule
You can set assessment generation schedules. You must set a schedule for each metric type individually.

Role required: assessment_admin or admin
1. Navigate to Assessments > Metric Definition > Types.
2. Open the appropriate metric type record.
3. Click the reference icon next to the Scheduled job field to open the Schedule Item record.
4. Locate the Trigger type field and set a recurring schedule for the scheduled job.

For example, if you want to generate assessment components each month, select the Interval or Repeat trigger type and specify a Repeat value of 30 days. Alternatively, select Day in Month and specify a Run time and Run day of month.

The optimal frequency of assessment generation is subjective. When you set the schedule, keep in mind the number of records to evaluate, how often your organization requires updated assessment results, the time and effort required of users to complete assessments, and other relevant factors.

Publish a metric type
Before it is possible to generate assessments, an assessment administrator must publish the associated metric type.

Role required: assessment_admin or admin

The State field on the Assessment Metric Type form indicates whether the type is published. The default state is Draft, which disables the system from generating assessments.

1. Navigate to Assessments > Metric Definition > Types.
2. Open a metric type record, and click Publish.
The **Publish** button is available only if the state is **Draft** and there is at least one metric associated with the type.

**Note:** A metric type cannot be moved back to the **Draft** state after it has been published.

Scheduled assessments

The system generates a unique scheduled job for each metric type with the **Schedule type** field set to **Scheduled**.

The system generates a unique scheduled job for each metric type with the **Schedule type** field set to **Scheduled**. Each scheduled job generates assessment components for the related metric type. By default, the scheduled job runs when an administrator executes it manually, but administrators can set a schedule to generate assessments automatically on a recurring basis.

For the system to properly generate scheduled assessments, the metric type must be active and **published**. The metric type must also be associated to at least one metric category. That metric category must be associated to one or more of each of the following items:

- **Assessable record**
- **Stakeholder** associated to one of the assessable records
- **Metric**

Generate a scheduled assessment manually

Administrators can generate scheduled assessments manually.

Role required: assessment_admin or admin

Use this option, for example, if you have set a schedule but want to generate assessments before the next scheduled run date.

1. Navigate to **Assessments > Metric Definition > Types**.
2. Open the appropriate metric type record.
3. Click **Generate Assessments** to trigger the scheduled job immediately.

**Note:** Be careful to click **Generate Assessments**, not **Generate Assessable Records**.

Generate a vendor type assessment manually

The Vendor Performance feature provides a direct method of generating assessments for the Vendor metric type.

Role required: admin

1. Navigate to **Vendor Performance > Admin > Generate Assessments**.
2. Click **Generate Assessments** to execute the scheduled job for the **Vendor** type.

Clean up assessment data

The assessment process generates a considerable amount of data, some of which is not useful after a short time.

Role required: assessment_admin or admin

The assessment process generates a considerable amount of data, some of which is not useful after a short time. Assessments include a scheduled job called Remove Old Assessment Data that is available to administrators only. The scheduled job removes these items if they are more than a year old:

- Assessment instance questions
- Assessment instances
• Metric results

By default, the scheduled job is executed manually. Administrators can configure the scheduled job to run automatically on a recurring schedule.

1. Navigate to **System Scheduler > Scheduled Jobs > Scheduled Jobs**.
2. Open the **Remove Old Assessment Data** record.
3. On the Schedule Item form, set the **Trigger type** field to run the scheduled job on a recurring schedule.
   
   It is recommended to set the scheduled job to run on a weekly or monthly basis.
4. Optional: Click **Execute Now** to run the scheduled job.

**Assessment instances**

An assessment instance represents one occurrence of a questionnaire assigned to one user.

The system generates assessment instances only when the required conditions are met, as described in *Scheduled assessments* and *On-demand assessments*, and there are non-scripted metrics in at least one category.

When the system generates scheduled assessments for a metric type, each assessment instance contains questions about assessable records and categories related to the stakeholder to which it is assigned.

Example:

Recall that there can be multiple stakeholder records associated with one user record. Minh Leclare is a stakeholder for these items related to the **Vendor** metric type:

<table>
<thead>
<tr>
<th>Assessable record</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>User Satisfaction</td>
</tr>
<tr>
<td>Acme</td>
<td>User Satisfaction</td>
</tr>
<tr>
<td>Acme</td>
<td>Reliability</td>
</tr>
<tr>
<td>Cisco</td>
<td>Reliability</td>
</tr>
</tbody>
</table>

When the system generates a scheduled assessment, Minh is assigned one assessment instance. Minh must evaluate Amazon, Acme, and Cisco by answering questions from the categories for which she is a stakeholder. Assuming that there are three questions in the User Satisfaction category and six questions in Reliability, Minh’s questionnaire contains three questions about Amazon, nine questions about Acme, and six questions about Cisco.

When the system generates an on-demand assessment *for a specific assessable record*, the assessment instance contains questions about that assessable record and all its associated categories. When the system generates an on-demand assessment *for a metric type*, the assessment instance contains questions about all that metric type’s assessable records and their associated categories.

**View an assessment instance**

An assessment instance represents one occurrence of a questionnaire assigned to one user.

Role required: none

1. Navigate to **Assessments > Assessments > Assessment Instances**.
2. Click an assessment instance number to open the record.
3. View the record, which displays the following information.
Table 86: Assessment Instance form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Auto-generated record number.</td>
</tr>
<tr>
<td>Metric type</td>
<td>Metric type of this assessment.</td>
</tr>
<tr>
<td>Assessment group</td>
<td>Assessment group to which this assessment belongs.</td>
</tr>
<tr>
<td>Due date</td>
<td>Date by which the assessment instance must be completed. The system populates the due date from the value in the metric type Assessment duration field. The system generates email notifications related to the due date.</td>
</tr>
<tr>
<td>State</td>
<td>State of the assessment. The possible states are Ready to take, In progress, Complete, and Canceled.</td>
</tr>
<tr>
<td>Assigned to</td>
<td>User this assessment is assigned to. This field becomes read-only when the state is In progress, Complete, or Canceled.</td>
</tr>
<tr>
<td>Signature</td>
<td>Name of the signature record attached to this assessment. A signature requires that assessment recipients acknowledge that they have read any assertions attached to a questionnaire.</td>
</tr>
<tr>
<td>Signature result</td>
<td>Verification provided by the recipient when a signature is required. This value is either the recipient's full name from the User [sys_user] table or checked, indicating that the recipient acknowledged reading the assertion by selecting a check box.</td>
</tr>
</tbody>
</table>

4. Click **Take assessment** to **open the questionnaire**.

This button is available if all the following conditions are true for the assessment instance:

- It is **Assigned to** you.
- The **State** is **Ready to take** or **In progress**.
- The associated metric type is active.

*Configure trigger conditions for an assessment*

Trigger conditions specify when to send a particular assessment and who to send the assessment to.
Role required: assessment_admin or admin

1. Navigate to **Assessments > Admin > Trigger Conditions**.
2. Complete the fields as described in the table:

**Table 87: Trigger Condition form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment</td>
<td>[Required] Metric type to generate assessment instances for. Triggered assessments only work if the metric type is set to the <strong>On demand</strong> schedule type.</td>
</tr>
<tr>
<td>Table</td>
<td>[Required] Table to run the trigger condition on. You can select only tables in the current application scope. The table must have a connection to the assessable records for the selected <strong>Assessment</strong>. For example, you might want to send a vendor assessment when incidents close that are related to vendors you assess. In this case, select the Incident [incident] table. After you select a table, ensure there is at least one option for the <strong>Assessable Record Field</strong>. If there is not, select a different table.</td>
</tr>
<tr>
<td>User field</td>
<td>[Required] Field that stores the users you want to send the assessment. You can select any field, on the selected <strong>Table</strong> or on a related table, that references the User [sys_user] table. Use the tree picker to select a field.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Repeat Interval</td>
<td>Minimum period that must pass before the trigger condition can resend the assessment to the same user. For example, assume the repeat interval is set to 30 days. Even if the same user is eligible for multiple assessments from this trigger condition, the system can only send the user one assessment every 30 days.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that determines whether this trigger condition is active (selected).</td>
</tr>
<tr>
<td>Business rule</td>
<td>[Admin only] Business rule the system creates to monitor the selected table. When the condition is met, the business rule sends the assessment to the correct user. No configuration is necessary for this business rule.</td>
</tr>
<tr>
<td>Trigger randomly</td>
<td>Check box that determines whether to send the assessment to the appropriate user every time the condition is met (cleared) or only a percentage of the time (selected), as specified in the <strong>Probability (%)</strong> field.</td>
</tr>
<tr>
<td>Probability (%)</td>
<td>Probability out of 100 that the assessment will be sent each time the condition is met. For example, if the probability is set to 50, the system sends the assessment approximately 50% of the time the conditions are met, assuming there are no repeat interval restrictions. This field is visible and required only when the <strong>Trigger randomly</strong> check box is selected.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Assessable Record Field</strong></td>
<td>[Required] Field on the selected <strong>Table</strong> that determines which assessable record will be the subject of the assessment. Only appropriate reference fields on the selected <strong>Table</strong> are available to select. If there are no <strong>Assessable Record Field</strong> options available, you must select a different <strong>Table</strong>. If the selected <strong>Assessment</strong> evaluates records on the Company [core_company] table and the selected <strong>Table</strong> is Incident, the only <strong>Assessable Record Field</strong> options are fields on the Incident table that reference the Company table. For example, <strong>Company</strong> or, if vendor ticketing is enabled, <strong>Vendor</strong>. If you select <strong>Vendor</strong>, the trigger condition sends an assessment about the <strong>Vendor</strong> for the incident, assuming there is an assessable record for the associated company. The assessment contains questions from all metric categories associated with the assessable record. <strong>Note:</strong> This field is not available if the selected <strong>Assessment</strong> is a metric type used for surveys. See <strong>Survey trigger conditions</strong> on page 596.</td>
</tr>
<tr>
<td><strong>Related Field 1 – Related Field 4</strong></td>
<td>Field that contains a value you want to store for reporting purposes. You can pick any reference field on the selected <strong>Table</strong>. When the trigger condition generates an assessment instance, the system stores the value from the triggering record. Select up to four fields. For example, if you select the Incident table, you might select <strong>Caller</strong> and <strong>Vendor</strong> as related fields. That stores the caller and vendor associated with the incident as <strong>Related record 1</strong> and <strong>Related record 2</strong> in the assessment instance record. To view these fields, configure the Assessment Instance form.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Summary information to identify the trigger condition.</td>
</tr>
</tbody>
</table>
Assessment trigger conditions

A trigger condition tells the system who to send a designated assessment to and when, based on specific conditions.

With trigger conditions, assessment administrators can configure the system to generate assessments each time a specific action occurs, such as when an incident or change request closes. The trigger condition sends the assessment to specified users who are related to the triggering record, such as incident callers or change request assignees. You can choose to send the assessment every time the condition is met, or set a probability for the system to send the assessment at random when the condition is met.

Table 88: Trigger condition configuration

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>[Required] Condition builder that defines the criteria that must be true to send the assessment. For example, if you want to send an assessment whenever an incident closes, create the condition [State] [is] [Closed].</td>
</tr>
</tbody>
</table>

Trigger condition example

Modesto Scroggie is the caller on an incident that is assigned to Boris Catino, a service desk technician. Boris contacts a vendor to resolve Modesto’s issue and sets the Vendor field to Microsoft. When the incident closes, the system creates an assessment instance assigned to Boris so he can evaluate Microsoft as a vendor.

Because two related fields were selected as part of the trigger condition, the assessment instance stores the following information from the incident:

- **Related record 1**: User: Modesto Scroggie
- **Related record 2**: Company: Microsoft
**Note:** Even though the trigger condition is set to be triggered every time the conditions are met, the **Repeat Interval** setting ensures that Boris does not receive another assessment if another vendor incident assigned to him closes within 30 days of the first one.

**Enable manager notifications**

Users with the assessment_admin role can enable the Notify manager assessment is overdue email notification.

**Role required:** assessment_admin or admin

This notification sends emails to assessors’ managers when assessors do not complete their assigned assessments on time. For more information, see the table of assessment notifications. You must enable or disable this email notification separately for each metric type.

1. Navigate to **Assessments > Metric Definition > Types**.
2. Open a metric type.
3. Select the **Notify if overdue** check box.

To disable manager notifications, clear the check box.

4. Save the record.

**Note:** The assessor’s user record must have a manager specified in the **Manager** field to use this notification. You might need to configure the form to use this field.

**Assessment notifications**

In the Assessments application, you can configure the instance to send email notifications to remind assessors and their managers of assessment questionnaires that must be completed. An assessment workflow triggers the notifications. Users with the required roles, as detailed in each section, can administer different aspects of the notification system.

**Note:** Ensure that your system is configured to send and receive email notifications.

**Included Email Notifications**

The Assessments application includes some predefined email notifications. Administrators can view and customize the provided email notifications or create new ones.

1. Navigate to **System Policy > Email > Notifications**.
2. Locate the following email notifications, all of which run on the Assessment Instance [asmt_assessment_instance] table, and edit them as needed. For more information about editing or creating email notifications, see **Create an email notification** on page 2756.
### Table 89: Email notifications

<table>
<thead>
<tr>
<th>Email notification name</th>
<th>Purpose</th>
<th>Email subject</th>
<th>Email message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify assessment user</td>
<td>Notifies assessors when they are assigned assessment instances. The notification includes the metric type, the due date from the <strong>Due date</strong> field on the Assessment Instance form, basic instructions, and a link to the assessment instance.</td>
<td>Assessment notification: New assessment on <code>&lt;metric type name&gt;</code> - due by <code>&lt;assessment instance due date&gt;</code>&lt;br&gt;Example: Assessment notification: New assessment on Vendor - due by 2013-07-30</td>
<td>You have been designated as an assessment participant on <code>&lt;metric type name&gt;</code> due to your role in the organization and subject matter knowledge. As required by management, please follow the link below and complete the assessment by <code>&lt;assessment instance due date&gt;</code>. You can save your responses until you are ready to submit them. When you are satisfied with your responses, submit the assessment.&lt;br&gt;Click here to take your assessment: <code>&lt;Assessment instance number hyperlink&gt;</code>&lt;br&gt;To view your assessment queue at any time, sign in and navigate to <code>&lt;correct navigation path based on version&gt;</code>.</td>
</tr>
<tr>
<td>Email notification name</td>
<td>Purpose</td>
<td>Email subject</td>
<td>Email message</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Remind assessment user</td>
<td>Follow-up notification for assessors. Reminds assessors when half the assessment duration elapses for Ready to take or In progress assessment instances that are assigned to them. The assessment duration is the period that begins when the system generates an assessment instance and ends on the due date. The notification includes the metric type, the due date, basic instructions, and a link to the assessment instance.</td>
<td>Assessment reminder: Assessment on &lt;metric type name&gt; - due by &lt;assessment instance due date&gt;</td>
<td>You have been designated as an assessment participant on &lt;metric type name&gt; due to your role in the organization and subject matter knowledge. As required by management, please follow the link below and complete the assessment by &lt;assessment instance due date&gt;. You can save your responses until you are ready to submit them. When you are satisfied with your responses, submit the assessment. Click here to take your assessment: &lt;Assessment instance number hyperlink&gt; To view your assessment queue at any time, sign in and navigate to &lt;correct navigation path based on version&gt;.</td>
</tr>
</tbody>
</table>

Assessment reminder: Assessment on Vendor - due by 2013-07-30

Example: Assessment reminder: Assessment on Vendor - due by 2013-07-30
<table>
<thead>
<tr>
<th>Email notification name</th>
<th>Purpose</th>
<th>Email subject</th>
<th>Email message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify manager</td>
<td>Notifies assessors' managers when assessors do not complete their assigned assessment instances by the due date. The notification includes the assessor's name from the Assigned to field on the assessment instance, the metric type, the due date, and a link to the assessment instance. Managers must have the assessment_admin role to view the assessment instance. An assessment administrator must enable this notification separately for each metric type.</td>
<td>Assessment notification: &lt;Assigned user name&gt;'s assessment on &lt;metric type name&gt; is overdue.</td>
<td>&lt;Assigned user name&gt; has not completed an assessment on &lt;metric type name&gt; by the due date: &lt;assessment instance due date&gt;. You are being notified because you are specified as &lt;assigned user name&gt;'s manager. Please contact &lt;assigned user name&gt; and take appropriate action. It is important that users complete assessments, as the data is critical to our business process decisions. Click here to view the overdue assessment: &lt;Assessment instance number hyperlink&gt;</td>
</tr>
</tbody>
</table>

Assessment notification workflow

The system sends assessment notifications according to the Notify assessment user workflow. Users with the workflow_admin, workflow_creator, or workflow_publisher roles can view workflows.

1. To open the graphical workflow editor, navigate to Workflow > Workflow Editor.
2. In the activity menu, click the link to choose an existing workflow.

![Figure 119: Workflow Window](image)

3. Select Notify assessment user from the Workflow Versions list.

The workflow appears.
Figure 120: Notify Assessment User Workflow

View a metric result
Metric results contain values that represent an evaluated record's performance for a specific metric, based on a single evaluation from one user or from the execution of a script. Metric results contain actual values as well as calculated values. The system uses values from metric results to calculate category results.

Role required: assessment_admin or admin

If there is an active scripted metric when the assessment is generated, the system automatically produces a metric result for each associated assessable record. The system produces an additional metric result for each question a user answers on a completed assessment questionnaire. If multiple users complete questionnaires with the same questions on the same assessable records, the system produces metric results for each user's responses. For example, three users complete questionnaires that contain the same four questions about the Acme Corporation record. The system produces 12 metric results: one per user for each question.

1. Navigate to **Assessments > Results > Metric Results**.
2. Click the reference icon ( ) to open the metric result record.
3. View the Metric Result form.
   All fields on the form except **Updated** and **String value** are read-only.

### Table 90: Metric Result form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment group</td>
<td>Assessment group to which the metric result is associated.</td>
</tr>
<tr>
<td>Metric</td>
<td>Name of the metric to which these metric result values apply.</td>
</tr>
<tr>
<td>Data type</td>
<td>Data type of the metric.</td>
</tr>
<tr>
<td>Method</td>
<td>Method of the metric.</td>
</tr>
<tr>
<td>Updated</td>
<td>Date and time the metric result was last updated.</td>
</tr>
<tr>
<td>Source</td>
<td>Source record of the assessable record evaluated.</td>
</tr>
<tr>
<td>Assigned to</td>
<td>User who completed the assessment questionnaire. This field is blank when the method is <strong>Script</strong>.</td>
</tr>
<tr>
<td>Instance</td>
<td>Assessment instance completed by the Assigned to user. This field is blank when the method is <strong>Script</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Actual value</td>
<td>Unscaled value from a user response or script, depending on the method:&lt;br&gt;  • Assessment: Value obtained from the user response to the assessment instance question. The actual value is determined by the metric data type:&lt;br&gt;   • Checkbox: The actual value is 0 if the check box is cleared and 1 if it is selected.&lt;br&gt;   • Choice or Likert Scale: The actual value is equal to the Value of the metric definition associated with the chosen answer option.&lt;br&gt;   • Date, Date/Time, or String: The actual value is -1 to indicate that these data types do not contribute to category result calculations.&lt;br&gt;   • Template: The actual value is equal to the Value of the template definition associated with the chosen answer option.&lt;br&gt;   • Yes/No: The actual value is 0 if the response is No and 1 if it is Yes.&lt;br&gt;  • Script: Value the script placed in the actual_result variable.&lt;br&gt;  This field is hidden and left blank when the data type is Duration.</td>
</tr>
<tr>
<td>Duration value</td>
<td>Specific kind of actual value that is only applicable if the data type is Duration. The duration value is the value obtained by the script query from the actual_result parameter, such as the average duration of outages for a vendor.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Scaled value  | Appropriate value from the `scaled_result` variable in the associated metric script. The code in a scripted metric defines a scaling scheme for the actual values it obtains. For example, a scripted metric queries the CMDB to return the number of configuration items (CIs) for a vendor. The code includes this scaling scheme, where `actual_result` is the number of CIs, and `scaled_result` is a representation of the quantity of CIs:  

```java
if (actual_result < 1)  
scaled_result = 0;
else if (actual_result < 1000)  
scaled_result = 1;
else if (actual_result < 10000)  
scaled_result = 2;
else if (actual_result < 25000)  
scaled_result = 3;
else if (actual_result < 100000)  
scaled_result = 4;
else  
scaled_result = 5;
```

If the script detects 315 CIs for the vendor, the Actual value is 315 and the Scaled value is 1 for this metric result.  

**Note:** Although this field is visible only when the method is **Script**, the system populates the field with the Actual value when the method is **Assessment**, as the scaled value is used in the **Normalized value** calculation.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Normalized value | Adjusted value that accounts for weights, scale definition, minimum and maximum values, and other factors that impact the metric. The normalized value is calculated as follows:  
Normalized value = (Input Value - Min value defined in metric) / (Max value defined in metric - Min value defined in metric) * current metric weight / (sum of valid metric weight) * scale_factor  
This formula cannot be customized.  
As an example, calculate the normalized value for this metric: Please rate the competency of the technician. This metric has the following values:  
• Input value = 3  
• Minimum value = 1  
• Maximum value = 6  
• Current metric weight = 10  
• Number of responses = 6  
  • 4 of type=number  
  • 1 of type=yes/no  
  • 1 of type=string (invalid data type; value cannot be calculated)  
  • Valid metric weight of each response = 10  
  • Scale factor = 10  
Normalized value = (3 - 1) / (6 - 1) * 10 / (10 + 10 + 10 + 10 + 10) * 10 = 0.8  
Several data types are ignored because the values cannot be calculated.  
These invalid data types include string, date, and datetime. For reporting purposes, use the Metric Result [asmt_metric_result] table. |
| String value   | Value that displays the response as it appears on a questionnaire. In some cases this is the same as the Actual value, such as when the metric data type is Percentage.  
The string value is N/A for unanswered questions of certain data types. |

**Assessment results**

In the Assessments application, when the system processes completed questionnaires or gathers values returned from scripted metric queries, it generates assessment result records called metric and category results.
Assessment administrators can view assessment results.

The system calculates **metric results** first, normalizes the scores, and then calculates **category results**. Each metric and category result record stores data from one assessment group, for a particular category and source record evaluated. Specifically:

- Metric results: store data calculated from one evaluation of one metric.
- Category results: store data calculated from a weighted average of all metrics in the category.

**Note:** Although results data is available to view in the form of metric and category result records, the most meaningful way to view results is in **scorecards** and **decision matrixes**. These tools display only the most pertinent data in graphical, comparative formats.

**Excluded Responses**

The system creates metric results for responses but does not include them in category result calculations if any of the following are true:

- The user selects the **Not Applicable** answer option on a questionnaire.
- The user does not answer the question and the **data type** is not Checkbox.
- The question data type is one of the following: Date, Date/Time, or String.
- The metric result **Actual value**, **Scaled value**, and **Normalized value** fields are set to -1.

**Assessments overview module**

The assessment overview module is a type of homepage that displays various assessment gauges, such as results by category and assessments by state.

**Prerequisites**

You can view the overview page and refresh, add, delete, and rearrange gauges.

Role required: assessment_admin
For details about editing gauges on homepages, see *Add existing gauges to a homepage* on page 478.

To use the Assessments Overview module, navigate to **Assessments** > **Overview** and click elements within the gauges to obtain more information.

The available gauges are:

**Table 91: Assessments overview module gauges descriptions**

<table>
<thead>
<tr>
<th>Gauge</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessable Records by Type</td>
<td>Assessable Record</td>
</tr>
<tr>
<td>Assessment Instances by Assessment Group</td>
<td>Assessment Instance</td>
</tr>
<tr>
<td>Assessments by State</td>
<td>Assessment Instance</td>
</tr>
<tr>
<td>Answered Questions by Assigned User</td>
<td>Assessment Instance Question</td>
</tr>
<tr>
<td>Average Rating by Type</td>
<td>Assessment Category Result</td>
</tr>
<tr>
<td>Category Result Ratings by Category</td>
<td>Assessment Category Result</td>
</tr>
<tr>
<td>Metrics by Data Type</td>
<td>Assessment Metric</td>
</tr>
<tr>
<td>Stakeholders per Category</td>
<td>Assessment Stakeholders</td>
</tr>
<tr>
<td>Top Rated Items</td>
<td>Metric Result</td>
</tr>
<tr>
<td>Total Metrics by Metric Type</td>
<td>Assessment Metric</td>
</tr>
<tr>
<td>Unanswered Questions by Assigned User</td>
<td>Assessment Instance Question</td>
</tr>
</tbody>
</table>

**Assessment results calculation**

For every assessment group, there are assessment results.

Data calculated from an evaluation of one metric is classified as a metric result. Data calculated from the weighted average of all metric results in a category is classified as a category result. The system calculates some assessment results when the assessment is generated, and others when users complete questionnaires.

**View a bubble chart for assessments**

Assessment administrators can view bubble charts to compare the relative standing of assessable records in three metric categories.

Role required: assessment_admin or admin

1. Navigate to **Assessments** > **Admin** > **Bubble Charts**.
2. Click a **Name** to open the bubble chart record.
3. Under **Related Links**, click **View Bubble Chart**.

**Note:** Demand managers and vendor managers can view bubble charts through the Demand Management and Vendor Performance applications.

**Bubble charts for assessments**

A bubble chart is a dynamically updated graph that plots assessment results for multiple assessable records.

Assessment administrators can create and use bubble charts to compare the relative standing of assessable records in three metric categories. The X- and Y-axes each represent a different category. Assessable records are plotted on the chart as circular markers, or bubbles, which vary in size according to scores for the third category.
Create a bubble chart
Assessment administrators can create bubble charts to compare the relative standing of assessable records in three metric categories.

To create a bubble chart, navigate to **Assessments > Admin > Bubble Charts** and create a record (see table for field descriptions).

**Note:** For color fields, either HTML color names or hexadecimal (hex) values are acceptable. For hex values, the # character is optional. Values are not case-sensitive. For example, all of the following values are valid: LightGray, lightgray, #D3D3D3, d3d3d3.

![Bubble Chart form](image)

Figure 122: Bubble Chart form
# Table 92: Bubble Chart form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Descriptive name for the bubble chart. <strong>Bubble Chart</strong> is appended to the name when you view the bubble chart.</td>
</tr>
<tr>
<td>• Top left label</td>
<td>Label text for the top left, top right, bottom left, and bottom right quadrants, respectively. Choose labels that help interpret results. For example, a bubble chart that displays vendor assessment results might have the following quadrant labels, where vendors in the top right quadrant have the best scores:</td>
</tr>
<tr>
<td>• Top right label</td>
<td>• <strong>Top left label</strong>: Resource</td>
</tr>
<tr>
<td>• Bottom left label</td>
<td>• <strong>Top right label</strong>: Essential Partner</td>
</tr>
<tr>
<td>• Bottom right label</td>
<td>• <strong>Bottom left label</strong>: Poor Performer</td>
</tr>
<tr>
<td>• Top left color</td>
<td>• <strong>Bottom right label</strong>: Reevaluate</td>
</tr>
<tr>
<td>• Top right color</td>
<td>Border colors for the top left, top right, bottom left, and bottom right quadrants, respectively. Bubbles are the color of the quadrant they are in.</td>
</tr>
<tr>
<td>• Bottom left color</td>
<td></td>
</tr>
<tr>
<td>• Bottom right color</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Metric type associated with this bubble chart. Only results for assessable records of the selected metric type are plotted on the bubble chart.</td>
</tr>
<tr>
<td>• X-Axis label</td>
<td>Label text for the X-, Y-, and Z-axis categories of the bubble chart. These labels appear along the X- and Y-axes, if applicable, and in bubble score summary windows. The metric category name is usually a good label. If these fields are left empty, the bubble chart automatically displays the selected category names as the labels.</td>
</tr>
<tr>
<td>• Y-Axis label</td>
<td></td>
</tr>
<tr>
<td>• Z-Axis label</td>
<td></td>
</tr>
<tr>
<td>• Metric X category</td>
<td>Metric category each axis represents. Results for the Z-axis category determine the size of each bubble. The Z-axis should generally represent the most important category out of the three.</td>
</tr>
<tr>
<td>• Metric Y category</td>
<td></td>
</tr>
<tr>
<td>• Metric Z category</td>
<td></td>
</tr>
<tr>
<td>Quadrant label color</td>
<td>Color of the label text for the quadrant borders.</td>
</tr>
<tr>
<td>Default</td>
<td>Check box that enables (selected) or disables (cleared) the bubble chart as the default. There can be only one default bubble chart per metric type.</td>
</tr>
</tbody>
</table>

**Bubble chart components**

The X- and Y-axes each represent a different metric category. Assessable records are plotted on the chart as circular markers, or bubbles, which vary in size according to scores for the third category.
The bubble chart page has these components:

- **Bubble Chart**
  - **X- and Y-axes**: Each axis represents a metric category.
  - **Bubbles**: Each labeled bubble represents an average of category result data for an assessable record. Point to a bubble to view an assessable record score summary. Click a bubble or bubble label to view the scorecard for the assessable record.

- **Source record list**: List of all records defined by the Table and Condition fields for the associated metric type. Note that the bubble chart only plots records for which there are assessment results. The assessment results are actually associated with the assessable records for each of the listed source records.
## Vendor Quality Bubble Chart

The chart illustrates the quality of vendors based on reliability and essential partner alignment. Each vendor is represented by a bubble, with its size indicating the level of alignment and position on the X-axis indicating the reliability score.

### Table: Vendor Quality Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Street</th>
<th>City</th>
<th>Zip / Postal Code</th>
<th>Phone</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acer</td>
<td>333 W. San Carlos St., Sta. 1500</td>
<td>San Jose</td>
<td>95110</td>
<td>(408) 533-7709</td>
<td>2013-09-05 08:47:42</td>
</tr>
<tr>
<td>Adobe Systems</td>
<td>345 Park Avenue</td>
<td>San Jose</td>
<td>95110-2704</td>
<td>408-536-6000</td>
<td>2013-09-09 13:30:25</td>
</tr>
<tr>
<td>Adtran</td>
<td>901 Explorer Boulevard</td>
<td>Huntsville</td>
<td>35038</td>
<td>(256) 963-8000</td>
<td>2013-08-06 15:29:17</td>
</tr>
<tr>
<td>Altec</td>
<td>350 Ellis Street</td>
<td>Mountain View</td>
<td>94043</td>
<td>(650) 527-8000</td>
<td>2013-08-08 10:50:49</td>
</tr>
<tr>
<td>Amazon</td>
<td>1200 12th Ave, South, Sta. 1200</td>
<td>Seattle</td>
<td>98114-2734</td>
<td>(888) 216-1072</td>
<td>2013-05-07 10:18:47</td>
</tr>
<tr>
<td>America Online</td>
<td>770 Broadway</td>
<td>New York, NY</td>
<td>10003</td>
<td>(602) 827-6354</td>
<td>2013-05-08 10:54:37</td>
</tr>
<tr>
<td>Apache Software Foundation</td>
<td>2000 Post Oak Boulevard, Suite 100</td>
<td>Houston</td>
<td>77086-4400</td>
<td>(713) 296-6000</td>
<td>2013-05-08 11:00:57</td>
</tr>
<tr>
<td>Apple</td>
<td>1 Infinite Loop</td>
<td>Cupertino</td>
<td>95014</td>
<td>(408) 996-1010</td>
<td>2013-05-07 10:21:57</td>
</tr>
<tr>
<td>Asus</td>
<td>800 Corporate Way</td>
<td>Fremont</td>
<td>94539</td>
<td>(510) 739-3777</td>
<td>2013-05-09 13:35:52</td>
</tr>
</tbody>
</table>
Assessable record score summaries

Point to a bubble to display a score summary for the assessable record the bubble represents.

The summary displays the assessable record's average score for each category in the following order: X-axis category, Y-axis category, Z-axis category. The scores are based on data from the last 12 months.

![Bubble chart score summary](image)

Figure 124: Bubble chart score summary

View an assessment scorecard

Scorecards contain a header that displays the name of the source record, and a section that compares the ratings. The Vendor Performance application provides an additional section for vendor scorecards and installs demo data that includes sample data for vendor assessments.

Role required: assessment_admin or admin

1. Navigate to **Assessments > Assessable Records**.
2. Open a record.
3. Under **Related Links**, click **View Scorecard**.

Assessment scorecards

The Assessments application prepares printable scorecards. A scorecard displays easy-to-interpret assessment results, in which the current calculated ratings for an assessable record are compared to previous ratings or to the ratings of other records.

Users can examine ratings over time, compare ratings for one assessable record with all assessable records in a table, or compare the ratings of two assessable records. All ratings are averages for the time range selected. The system dynamically updates a scorecard each time you view it, so the ratings reflect recently completed assessments and scripted metrics.

Administrators can display a scorecard for any table by creating a **Related Link** on assessable records. The system provides this link on the Company form by default when Vendor Performance is active.

The Vendor Performance application provides an enhanced scorecard view for vendors.

Create a link to a scorecard

Users with the admin role can create UI actions that allow users to view scorecards from tables.
Role required: assessment_admin or admin

1. Generate **assessable records** you want to evaluate.
   For example, you might create a metric type called **Project** to assess project management records.

2. Navigate to **System Definition > UI Actions**.

3. Open the View Scorecard record.

4. Right-click the header bar and select **Insert and Stay** from the context menu to create a duplicate record.

5. Change the **Table** name to the table on which you want the UI action to appear.
   For example, you might select **Project [pm_project]**.

6. Do not edit the **Action name** field or the **Condition** script.

7. Save the record.

8. Navigate to the table on which you created the UI action and open an assessable record.
   a) In this example, navigate to **Project > Projects > All**.
   b) Open any record in the list.
   c) Click **View Scorecard** under **Related Links** to open the scorecard for that assessable record.
The scorecard appears with the title in the form of `<table display name> Scorecard`. For example, a scorecard for an assessable record in the `Project [pm_project]` table is named `Project Scorecard`.

**Note:** Content does not appear in the scorecard unless the associated assessable record has assessment results or related live feed conversations.

9. Insert a new `View Scorecard` UI action record for each table where you want the related link to appear.

Export a scorecard as an image

You can export scorecards as images.

Role required: assessment_admin or admin

1. Click the menu icon and select `Save as PNG` or `Save as JPEG` as the download format.
2. When the export is complete, select `Download` to save the scorecard image to a storage location.

Averages

The `Averages` view compares the current ratings for an assessable record in each metric category with the average, minimum, and maximum values from all assessable records in the filter. All ratings are from assessments generated over the trailing twelve months (TTM).
Select a filter option to compare the current record against all assessable records in that filter. An assessment administrator configures these options in the Filter field field in the Assessment Metric Type form.

Rating variances are highlighted as follows:

- **Red**: Ratings are below average in this category. The Diff value displays a negative number.
- **Green**: Ratings are above average in this category. The Diff value displays a positive number.
- **White**: Ratings are average in this category. The Diff value is 0.0.

In the following example, the ratings of the current group are compared against other groups for which Don Goodliffe is the manager.

**Figure 125: Group scorecard with ratings filter**

Scorecard categories
The Categories view displays a bar chart showing the average ratings for each category in the selected time interval.

Select from these reporting periods:
- All History
- Last 3 Months
- Last 12 Months
Figure 126: Scorecard categories

Scorecard category metrics

The Category Metrics view displays the weighted average results for each metric within a category. Use this view to learn how individual metrics affect the overall rating for the category.

Select a metric category from the choice list to display the chart.
Figure 127: Scorecard category metrics

Head to head compare
The Head to Head Compare view allows you to compare the ratings of two assessable records of the same type. Select an assessable record from the choice list to compare against the current record's trailing twelve month (TTM) ratings.

Head to head compare
The Diff column displays the difference between each assessable record's most recent TTM ratings. By default, the system selects the first assessable record in the list when you open this view. The scorecard displays three years of ratings for the comparison record. All ratings are expressed as averages.
Figure 128: Scorecard head to head compare

**Overall Rating**

The Overall Rating is calculated as:

\[
\text{(sum of normalized values in category result)} / \text{(number of assessment groups)}
\]

In the following example, the calculation is

\[
(2.13 + 2.86 + 3.79 + 1.43 + 2.39 + 3.7) / 2 = 8.15
\]
<table>
<thead>
<tr>
<th>Assessment group</th>
<th>Category</th>
<th>Weight</th>
<th>Source</th>
<th>Rating</th>
<th>Normalized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASG0010010</td>
<td>Politeness</td>
<td>5</td>
<td>Group: NY DB</td>
<td>9.37</td>
<td>2.13</td>
</tr>
<tr>
<td>ASG0010010</td>
<td>Recommendation</td>
<td>7</td>
<td>Group: NY DB</td>
<td>9</td>
<td>2.86</td>
</tr>
<tr>
<td>ASG0010010</td>
<td>Responsiveness</td>
<td>10</td>
<td>Group: NY DB</td>
<td>8.34</td>
<td>3.79</td>
</tr>
<tr>
<td>ASG0000804</td>
<td>Politeness</td>
<td>5</td>
<td>Group: NY DB</td>
<td>6.29</td>
<td>1.43</td>
</tr>
<tr>
<td>ASG0000804</td>
<td>Recommendation</td>
<td>7</td>
<td>Group: NY DB</td>
<td>7.5</td>
<td>2.39</td>
</tr>
<tr>
<td>ASG0000804</td>
<td>Responsiveness</td>
<td>10</td>
<td>Group: NY DB</td>
<td>8.14</td>
<td>3.7</td>
</tr>
</tbody>
</table>

**Figure 129: Normalized values**
### Group Scorecard

<table>
<thead>
<tr>
<th></th>
<th>Current</th>
<th>Diff</th>
<th>2014</th>
<th>2013</th>
<th>2012</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overall Rating</strong></td>
<td>8.15</td>
<td>3.63</td>
<td>4.52</td>
<td>6.48</td>
<td>9.11</td>
</tr>
<tr>
<td><strong>Politeness</strong></td>
<td>7.83</td>
<td>3.68</td>
<td>4.15</td>
<td>5.57</td>
<td>8.86</td>
</tr>
<tr>
<td><strong>Recommendation</strong></td>
<td>8.25</td>
<td>5.00</td>
<td>3.25</td>
<td>6.00</td>
<td>9.25</td>
</tr>
<tr>
<td><strong>Responsiveness</strong></td>
<td>8.24</td>
<td>2.63</td>
<td>5.61</td>
<td>7.25</td>
<td>9.16</td>
</tr>
</tbody>
</table>

**Figure 130: Overall Rating on the Group Scorecard**

**History**

The History view compares the current ratings for each category with ratings from the previous three years or four quarters.

Ratings that have declined are highlighted in red and display negative numbers. Ratings that have improved are highlighted in green with positive numbers. Arrow icons beside the values in the **Diff** column indicate the trend of the current assessment against the previous assessment.

- **3 Years**: To calculate the current ratings, the system averages the ratings from the trailing twelve month (TTM) period. The **Diff** column shows the discrepancy between the current ratings and the previous calendar year's ratings.
Figure 131: Scorecard history - 3 years

- **4 Quarters**: Quarterly assessments compare the average rating for each category in the current quarter against the average ratings from the previous four quarters. The Diff column shows the discrepancy between the current ratings and the previous quarter's ratings. The column labels count backward, by quarter from the current quarter. For example, if the current quarter is the 3rd quarter of 2013, then the previous quarters appear as 2nd [2013], 1st [2013], 4th [2012], and 3rd [2012]. All four of the previous quarters appear, whether or not there was any data for those quarters.

Figure 132: Scorecard history - 4 quarters

Live feed

The live feed view displays live feed conversations that apply to the assessable record. Users can view the group feed, including tagged comments, and add comments of their own.
To configure a live feed view, an assessment administrator selects the **Live feed** check box on the **Assessable Record** form. Live feed is available on the scorecard even if there are no category results for the other views.

The live feed view provides these viewing options:

- **My Feed**: Displays the user’s My Feed preferences.
- **Company**: Displays the user’s company feed.
- **Groups**: Displays all available groups and gives the logged in user the option to leave or join a group. Select a group to display that group's conversations.
- **Tags**: Displays all the tags found in live feed comments. Click a tag to display the conversations containing that tag.
Figure 133: Scorecard Live Feed group
The scorecard ratings section displays various ratings for the assessable record.

Select one of several views that present different ratings:

- Averages
- Categories
- Category Metrics
- Head to Head Compare
- History
- Live Feed

Some views display an overall rating column, which lists categories used to evaluate the assessable record. Only categories in which the assessable record has been evaluated appear on the scorecard. If the ratings section does not display any data, the assessable record or category results associated to the assessable record have been deleted.

Click a category to view the category record. Point to a category to display a line chart that shows the rating trend for that category.

Set up and administer quizzes

Set up and administer quizzes.

Role required: assessment_admin or admin

1. Optional: Create reusable question templates for quizzes.
2. Create the quiz using the Quiz Designer or with forms accessed from the navigation menu.
   The quiz record includes specifics such as duration, notification preferences, a questionnaire introduction, and ending notes displayed to recipients.
3. Edit the default category or create additional categories as needed.
   The system creates a category with the same name as the quiz.
4. Define users for each category.
   These are the recipients who answer the questions in a category. You can define different users for each category.
5. Create the questions for each category.
6. Create the answers for each question and determine if the questions are scored.
You can create unique answers or select preconfigured answers from a template.

Publish and distribute the quiz. You can send the quiz to a single user or all users in each category.

Review the results from the submitted quizzes in reports and scorecards.

**Quizzes**

Quizzes are questionnaires you can assign to one or more users to assess their knowledge of any subject. The quiz functionality is built on the assessment engine and provides many of the same features as assessments and surveys.

Each question is scored, and the overall score indicates the percentage of questions the user answered correctly. A quiz may have categories of questions that are assigned only to some users. You can assign weighting values to individual questions or categories of questions that make them more or less important when calculating the overall score. Quizzes require activation by a system administrator.

- An administrator can create a quiz for any purpose and assign it to a single user or multiple users.
- A quiz can contain one or more categories of questions. Each category can be assigned to users who answer only the questions in that category.
- The system can send email notifications to these users:
  - Recipients: The recipient can receive notification of an assigned quiz, a quiz whose allowed duration is at 50%, and a quiz that is overdue.
  - Recipient's manager: The recipient's manager can receive notification when a quiz is overdue.
  - Quiz manager: The quiz manager can receive notification of an overdue quiz to which he or she is assigned.

- Quizzes can contain questions that are scored or not scored. Unscored questions assess opinions or involve dates and are not counted in the final score. Scored questions specify correct answers and are scored either as 0% or 100%. You can apply a weighting scale to scored questions to establish their relative importance. You can designate questions with these data types as scored questions:
  - Checkbox
  - Choice
  - Duration
  - Likert Scale
  - Numeric Scale
  - Template
  - Yes/No

- A quiz question can be dependent on the response to any scored question. For example, you can create a dependent question requesting additional information that appears only if a recipient answers No to a specific question.

**Data types for quizzes**

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkbox</td>
<td>Check box to use as a response to a True/False question.</td>
</tr>
<tr>
<td>Choice</td>
<td>List of predefined options. When you select <strong>Choice</strong>, answer option fields. You must enter at least two answer options.</td>
</tr>
<tr>
<td>Date</td>
<td>Date field</td>
</tr>
<tr>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Time and date field.</td>
</tr>
<tr>
<td>DurationPercentage</td>
<td>Percentage field with a prescribed range. When the script runs, the system</td>
</tr>
<tr>
<td></td>
<td>populates the <strong>Duration value</strong> and <strong>Scaled value</strong> fields on the Metric</td>
</tr>
<tr>
<td></td>
<td>Result form with the appropriate values from the actual_result and scaled_</td>
</tr>
<tr>
<td></td>
<td>result variables in the metric script.</td>
</tr>
<tr>
<td>Likert Scale</td>
<td>Predefined <strong>Likert scale</strong> of answer options, each represented by a radio</td>
</tr>
<tr>
<td></td>
<td>button on the scale. When you select Likert Scale, a field appears for you</td>
</tr>
<tr>
<td></td>
<td>to enter the answer option text. When you make an entry in this field,</td>
</tr>
<tr>
<td></td>
<td>another field appears below it for the next answer option on the scale. You</td>
</tr>
<tr>
<td></td>
<td>must enter at least two answer options.</td>
</tr>
<tr>
<td>Number</td>
<td>Number field with predefined minimum and maximum values.</td>
</tr>
<tr>
<td>String</td>
<td>Single or multi-line text field.</td>
</tr>
<tr>
<td>Template</td>
<td>Choice list of templates that provide a predefined scale of options.</td>
</tr>
<tr>
<td>Yes/No</td>
<td>Selections for a response to a True/False question.</td>
</tr>
</tbody>
</table>

**Quiz roles**

The Quizzes application uses these roles. No role is required to take quizzes that are assigned to you.

### Table 94: Quiz roles

<table>
<thead>
<tr>
<th>Role Title [Name]</th>
<th>Role Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assessment administrator [assessment_admin]</td>
<td>Can administer the Assessments application and all quiz records. Can access all the modules of the Assessments application.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The itil_admin role and the survey_admin role contain the</td>
</tr>
<tr>
<td></td>
<td>assessment_admin role</td>
</tr>
<tr>
<td>Administrator [admin]</td>
<td>Can access all aspects of the assessment and survey processes. Only administrators</td>
</tr>
<tr>
<td></td>
<td>can modify survey notifications, create survey modules, and import surveys.</td>
</tr>
</tbody>
</table>

**Quiz application: Important terms**

The quiz application involves several terms.
Table 95: Terms used in quiz application

<table>
<thead>
<tr>
<th>Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes</td>
<td>A quiz contains information such as duration, state, and notification controls and lists the existing categories. Text fields on the quiz form allow an administrator to create introductory content and end notes that are displayed to the recipient.</td>
</tr>
<tr>
<td>Categories</td>
<td>A quiz category represents a theme for quiz questions. Each category contains one or more questions and names the recipients for the questions in that category. By default, the system creates one category with the same name as the quiz. You can create additional categories as needed. Categories can be weighted higher or lower to determine the importance of that category in the overall score.</td>
</tr>
<tr>
<td>Questions</td>
<td>A quiz question is a question configured for a category and sent only to the users for that category. Questions have a wide variety of data types and can be individually weighted higher or lower. Questions may be scored or unscored.</td>
</tr>
<tr>
<td>Category user</td>
<td>A category user is the recipient of questions for a specific category. You can select different users to answer the questions for each category.</td>
</tr>
<tr>
<td>Templates</td>
<td>A template is a question data type that provides reusable rating scales for answers to questions. For example, the answer template named Satisfaction contains a satisfaction scale ranging from Very Satisfied to Very Dissatisfied.</td>
</tr>
</tbody>
</table>

Activate the quiz designer

Administrators can activate the Quiz Designer plugin.

Role required: assessment_admin or admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Quiz Overview module

The Quiz Overview module is a homepage that displays various gauges that report on data such as results for each category and quizzes that are complete, pending, or in progress.

Role required: assessment_admin or admin
For details about editing gauges on homepages, see *Add existing gauges to a homepage* on page 478.

Users with the `assessment_admin` role can view the overview page and refresh, add, delete, and rearrange gauges.

1. Navigate to **Quizzes > Overview**.
2. Click elements within the gauges to obtain more information. The available gauges are:

<table>
<thead>
<tr>
<th>Gauge</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quizzes by State</strong></td>
<td>Assessment Instance [asmt_assessment_instance]</td>
</tr>
<tr>
<td><strong>Total Questions by Quiz</strong></td>
<td>Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td><strong>Questions by Data Type</strong></td>
<td>Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td><strong>Correct Answers by Assigned User</strong></td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td><strong>Number of Correct Answers</strong></td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td><strong>Incorrect Answers by Assigned User</strong></td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td><strong>Number of Incorrect Answers</strong></td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
</tbody>
</table>

**Quiz designer**

The quiz designer provides a single interface that users with the `assessment_admin` role can use to create, edit, and distribute quizzes.

You can also use it to edit existing quizzes and change scoring parameters.

Alternatively, you can use the modules of the *assessment engine* to create and edit the records that make up a quiz. All quiz records are stored in assessment tables and displayed in Quiz views of those tables.

**Tools on the Quiz Designer**

The quiz designer includes a design canvas, a header bar, and many controls that you can use to create quizzes.

To open the quiz designer, navigate to **Quizzes > Quiz Designer**.

The designer contains the following elements:

- Controls
- Header bar
- Design canvas

**Controls**

Controls for the supported question data types are available in the Controls palette. Drag and drop a control onto the designer canvas to create a question of that type.
# Question controls

**Figure 135: Question controls**

### Table 96: Question controls

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
<th>Scored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Question with a check box or a <strong>Yes/No</strong> list for user responses.</td>
<td>Y</td>
</tr>
<tr>
<td>Choice</td>
<td>List of predefined options. For more information, see the definition for <strong>Choices Create quiz questions</strong> on page 403.</td>
<td>Y</td>
</tr>
<tr>
<td>Date</td>
<td>Date field.</td>
<td>N</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Date and time field.</td>
<td>N</td>
</tr>
<tr>
<td>Number</td>
<td>Number field with predefined minimum and maximum values. The default is 1-10.</td>
<td>N</td>
</tr>
<tr>
<td>Percentage</td>
<td>Percentage field with a prescribed range.</td>
<td>N</td>
</tr>
<tr>
<td>Data type</td>
<td>Description</td>
<td>Scored</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Scale</td>
<td>Predefined <em>Likert scale</em>. Answer options appear as radio buttons.</td>
<td>Y</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>Selectable number scale. The default is 1-5. Answer options appear as radio buttons.</td>
<td>Y</td>
</tr>
<tr>
<td>String</td>
<td>Single or multiline text field.</td>
<td>N</td>
</tr>
<tr>
<td>Template</td>
<td>Choice list of templates that provide a predefined scale of options. For details, see <em>Configure a template question</em>.</td>
<td>Y</td>
</tr>
</tbody>
</table>

**Header Bar**

The header bar contains tabs that display different views and a menu of various functions.

Click one of the following tabs to change the view in the canvas:

- **Design**: Add categories and questions, and configure the properties of each. This is the default view of the canvas when you open the designer.
- **Configuration**: Create introductions and end notes for quizzes, and select a signature.
- **Availability**: Select the recipients for each category in the quiz.

Point to the menu icon (≡) in the in the upper right of the quiz designer to select the following options:

- **Save**: Saves the current quiz.
- **Preview**: Displays a preview of the quiz as it appears to the recipients.
- **Publish**: Distributes the quiz to the selected recipients.
- **Save and Publish**: Saves and distributes the quiz in one step.
- **New Quiz**: Opens a fresh canvas for a new quiz.
- **Load Quiz**: Opens a list of existing quizzes that you can select and edit.

The availability of each option depends on the status of the quiz that is opened in the quiz designer.

**Design Canvas**

New quizzes open in the **Design** view. The quiz **Name** field appears above first category in the canvas. A blank question field appears in the category container.

Create a quiz

When you create a quiz, you can create one or more categories and then add questions to each category.

Role required: assessment_admin or admin

Each category can be assigned to a different user or the same users. You can also customize each question and make it dependent on the response to another question.

Create a quiz using these procedures:

Create quiz categories

A category represents a theme for evaluating a specific element of the quiz topic and contains questions pertaining to that theme.

Role required: assessment_admin or admin
When you create a quiz, the system creates a default category, using the name of the quiz. You can use this category, modify it, or create additional categories as needed. To have any results, a category must contain scored questions.

1. Navigate either Quizzes > Quiz Designer or Quizzes > Quizzes and click Quiz Designer in the list header.
2. Enter the name of the quiz in the Name field. The system uses this name as the name of the quiz and the first category.
3. To configure the category, click the gear icon in its title bar. The Properties dialog box appears. You can change the name of the category, add a description for it, and enter text in the Details field that introduces or explains the category to the recipients. The system updates the category as you type.
4. Click the X icon to close the category properties dialog box and save your settings.

5. To add a new category, click the + icon in the title bar of an existing category.

Create quiz questions
You can create multiple questions for each category but each question can be associated with only one category.

Role required: assessment_admin or admin

The data type that you select for each question determines how it can be answered by quiz recipients.

You can designate questions to be scored. Only scored questions are shown in the quiz results and considered when calculating the category results. You must also specify a correct answer for scored questions.
Note:
To designate a question as scored, you must use Assessment forms. For instructions, see *Configure a scored question*.

1. In the Design view, drag a data type icon from the Controls palette and drop it into a category container.

2. To configure the question, click in the gear icon in its title bar. The Properties dialog box appears for the question.
3. Fill in the fields on the form, as appropriate.

Table 97: Question property fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Required] Name of the question. Create a concise and easily recognizable name for your question. The system uses this value to identify the question in Assessment Metric lists and in scorecard charts.</td>
</tr>
<tr>
<td>Question</td>
<td>Text to display as the question on quizzes. Enter a clear, straightforward question that is easy to understand.</td>
</tr>
<tr>
<td>Type</td>
<td>[Read-only] Data type selected for this question. See the table in Controls for possible data types.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that determines whether this question is available on a quiz. If a question is marked inactive, it does not appear on quizzes generated after the question becomes inactive.</td>
</tr>
<tr>
<td>Boolean option</td>
<td>Whether a check box or a Yes/No list appears as the option for the Boolean question.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Check box for requiring users to answer the question. Users cannot submit quizzes until they answer all mandatory questions, which are denoted by a red field status indicator. This field is available when the question does not have a dependency and the question Controls is not Boolean with a check box option.</td>
</tr>
</tbody>
</table>
| String option     | Setting for the appearance of a string field in a question. This field is available when the question type is String. The string options are:  
  - **Single line**: Single line text field 40 characters in length that allows strings of any length.  
  - **String line wide**: Full page width text field that allows a single line entry of any length.  
  - **Multiline**: Full page width multi-line text field that allows word wrap and returns. |
<p>| Min               | Lowest positive whole number that users can enter or select to answer the question. This field is available when the question type is Number, Percentage, or Numeric Scale. |
| Max               | Highest positive whole number that users can enter or select to answer the question. This field is available when the question type is Number, Percentage, or Numeric Scale. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow not applicable</td>
<td>Check box for including <strong>Not Applicable</strong> as an option for this question. Users can select <strong>Not Applicable</strong> if they do not have sufficient information to respond to a question. User responses of <strong>Not Applicable</strong> are excluded from results calculations. This field is available when the question does not have a dependency and the question <strong>Date type</strong> is not <strong>Boolean</strong> with a check box option.</td>
</tr>
</tbody>
</table>
| Randomize answers          | Check box for displaying answer options in a random order whenever the question appears in a quiz. Answer preference is sometimes influenced by the order in which options appear, which can result in biased results. Randomizing options can help prevent this bias.  
**Note:** Randomizing options for certain questions may make those questions confusing for users. In general, only randomize options that do not follow a logical order. |
| Details                    | Information about the question that is displayed on the quiz. Include details that help users understand how to answer the question. You can also enter HTML text in this field with the WYSIWYG editor. For example, include HTML to embed links and images. |
| Correct answer             | Answer option that you want to be selected by users. When you specify a correct answer for a question, the system scores the question. This field is available for all data types except **Date**, **Date/Time**, and **String**. |
### Field | Description
--- | ---
Choices | Options for a question with a data type of **Choice** or **Scale**. The system automatically adds text and values that you can edit for each option. Click the + icon to add an option, or click the X icon to delete an option. By default, the system arranges options in the order established by their values. To change the order, drag and drop the options.

**Dependency**

**Displayed when** | Condition builder that hides or displays the question depending on the answer to another question in the same category. Select an existing question from the list with a data type of **Boolean**, **Choice**, **Scale**, or **Template**. Create the condition that must exist for recipients to see the dependent question, using the is or is one of operator. The system prevents recursive dependencies between questions. For example, if Question A depends on Question B, Question B cannot depend on Question A.

4. To create any special conditions that must be met for a question to appear on the quiz, click the **Dependency** tab. Fill in the fields, as appropriate.

Dependent questions appear on the quiz when a recipient selects a specific answer or combination of answers to another question in the same category

5. Select a question in the Displayed when field. The system selects the appropriate operator and displays the possible answers for the selected question.

6. Select the answers that satisfy the condition. Selected answers are indicated by a check mark.
7. Click X to close the question properties dialog box and save your settings.
8. To add a question with the same data type as an existing question in the category, click the + in the title bar of the existing question.
9. Drag questions to change their order within a category or move them between categories.
10. To delete a question, click the X in its title bar.

Configure a quiz
You can configure an entire quiz.
Role required: assessment_admin or admin
The values you enter and select on this page are applied to the entire quiz.

In the quiz designer, click Configuration and then fill in the fields as described in the table.

Table 98: Quiz designer configuration

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Check box for enabling the distribution of this quiz to recipients.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of this configuration or the quiz to which it is attached.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Introduction</td>
<td>Introductory content to display on quizzes. You can add a welcome message or background information about the quiz.</td>
</tr>
<tr>
<td>Signature</td>
<td>[Optional] Acknowledgement by a quiz recipient of requirements, admonitions, or expectations related to a quiz.</td>
</tr>
<tr>
<td>Return URL</td>
<td>Destination address of a web page that is presented to users after they submit a completed quiz. When a return URL is configured, the End note content does not appear.</td>
</tr>
<tr>
<td>End note</td>
<td>Content that is displayed to recipients after they submit a completed quiz. You can add a thank you message, follow-up instructions, or other applicable information. End notes are not displayed if a Return URL is specified.</td>
</tr>
<tr>
<td>Duration</td>
<td>Amount of time that recipients are given to complete this quiz, starting from the time that the quiz is generated. The default duration is 14 days.</td>
</tr>
<tr>
<td>Manager</td>
<td>Assessment manager for this quiz. These users are only responsible for managing the quiz process and not the results. The system notifies the manager when submissions for this quiz are past due.</td>
</tr>
<tr>
<td>Notify manager if overdue</td>
<td>Check box for sending email notifications to a recipient's manager when that user fails to submit an assigned quiz before the due date.</td>
</tr>
</tbody>
</table>

Select a quiz recipient

When the system distributes a quiz, it sends email notifications to the category users and their managers.

Role required: assessment_admin or admin

A category can have one or more assigned users, and the same user can be assigned to more than one category.

The system also creates a link to the quiz in the recipients’ My Assessments & Surveys portal. Users can only answer questions in the categories that they are assigned to.

To select the recipients for each quiz category, click Availability and then fill in the fields as described in the table.

### Table 99: Select recipients

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select category</td>
<td>Category for which the selected users are recipients.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Add users</td>
<td>Users selected as recipients for this category. The choice list shows all users in the system.</td>
</tr>
</tbody>
</table>

Publish a quiz
When you publish a quiz, the system sends email notifications to the recipients and to their managers. A card in each recipient's My Assessments & Surveys portal displays a link to the quiz.

Role required: assessment_admin or admin

1. If you want to see the quiz as the recipients will see it, point to the menu icon in the upper right of the quiz designer and click Preview.
2. When you are satisfied with the quiz, click Save and Publish or Publish to distribute it.

Create quizzes with forms
As an alternative to the Quiz Designer, you can create a complete quiz using records in the Assessment application.

All the elements of a quiz, the categories, questions, and answers, are stored in tables used by the assessment engine and are displayed in quiz views of these tables. Users creating quizzes in the Assessment application must have the assessment_admin role.

The best practice for creating a quiz using assessment forms is to follow the procedures in the order shown here:

- Create the quiz.
- Set up the categories.
- Create the questions for the quiz.
- Create the answers for the questions.
- Distribute the quizzes to recipients.

**Note:** The recommended method of creating and editing quizzes is to use the quiz designer, which provides a single, intuitive interface for creating and editing quizzes quickly. If you determine that you need to add specific features to your quiz not offered through the quiz designer, you can do so by using some of the specific procedures described here.

Set up a category
A category represents a theme for evaluating a specific element of the quiz topic and contains questions pertaining to that theme.

When you create a quiz, the system creates a default category with the name of the quiz. You can use this category, modify it, or create additional categories as needed. Categories are records in the Quiz view of the Metric Category table.

Categories have weighting values that contribute to the overall score for the quiz. By default, all categories are given a weighting value of 10. You can assign any weight to your categories. To have any results, a category must contain scored questions.

You can create a new category or edit an existing one from the Metric Categories related list of a quiz record.
Select a user for a category

Category users are the recipients of the questions for each category.

A category can have one or more assigned users, and the same user can be assigned to more than one category. When the system distributes a quiz, it sends an email notification, if configured, to the category users and creates a link to the appropriate questionnaire in their assessments and surveys portal. Users can only answer questions attached to their assigned categories.

To select users for a category, select the **Users** related list in a category record, and click **Edit**.
Create questions

A category can have multiple questions associated with it.

Each question can only be associated with one category. Each question has an answer data type that determines how recipients answer the question. Questions are records in the Quiz view of the Assessment Metric table.

To create questions, open a category record and create a question record from the Assessment Metrics related list. Alternatively, you can navigate to Quiz Management > Questions and create a new record (see table).

**Warning:** When you create a Choice or Likert Scale question, you must reopen the Assessment Metric form after you submit it to create answers. If you distribute a questionnaire without creating the answers for questions with these data types, recipients are unable to answer the questions. If the questions are mandatory, the recipients are unable to submit their questionnaires.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Required] Name of the question.</td>
</tr>
<tr>
<td>Category</td>
<td>[Required] Category the question belongs to. The system populates this category if you create a new question from the Metric Category form.</td>
</tr>
</tbody>
</table>

**Note:** You cannot change the category if the **Depends on** field is set or if another metric depends on this metric.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Method  | Setting that determines how to use the question.  
- **Assessment**: Makes the question available on a quiz distributed to users. The **Assessment** method is compatible with all data types except **Duration**.  
- **Script**: Queries the database without user participation. Scripted questions are of limited value for quizzes, because they do not assess a user's knowledge of a topic.  
**Note**: If you select a **Data type** that is incompatible with the selected **Method**, the system automatically changes the **Method** to the correct value. |
| Weight  | Numeric value that represents the importance of this question relative to other questions in the same category. By default, the weight is 10. For more information on weight values, see *Weight categories and metrics* on page 324. This field is available and required unless the **Data type** is **Date**, **Date/Time**, or **String**. These data types are not included in results calculations. |
| Order   | [Required] Numeric value that determines the order of the question in the category. The question with the smallest order value appears as the first question in the category’s section. By default, the order is 100.  
**Note**: It does not matter which order value you use for metrics with the **Script** method, because they do not appear on questionnaires. |
<p>| Active  | Check box that determines whether this question appears on quizzes. If a question is inactive, it does not appear on quizzes generated after the question becomes inactive. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Check box for requiring users to answer the question. Users cannot submit questionnaires until they provide valid responses to all mandatory questions, which are denoted by a red field status indicator. This field is available only if the Method is Assessment, the Depends on field is empty, and the Data type is not Checkbox.</td>
</tr>
<tr>
<td>Allow not applicable</td>
<td>Check box for including Not Applicable as a possible answer for this question. Users can select Not Applicable if they do not have sufficient information to respond to a question. User responses of Not Applicable are excluded from results.</td>
</tr>
<tr>
<td>Question</td>
<td>The question that is displayed in quizzes. Enter a clear, straightforward question that is easy to understand.</td>
</tr>
<tr>
<td>Details</td>
<td>Information about the question and what it evaluates. Include details that help users understand how to answer the question or when you need HTML enhanced details for your question. You can create HTML text in this field with the WYSIWYG editor, such as embedding links and images.</td>
</tr>
<tr>
<td>Depends on</td>
<td>An existing question that this question is dependent on. You can select Checkbox, Choice, Likert Scale, Template, and Yes/No questions from the same category as this question. Then use the Displayed when field to set the conditions for displaying this question. The system prevents the creation of recursive dependencies between questions. For example, if Question A depends on Question B, Question B cannot depend on Question A.</td>
</tr>
<tr>
<td>Question Type</td>
<td></td>
</tr>
<tr>
<td>Data type</td>
<td>[Required] Format of the expected response data. See the table of data types for details.</td>
</tr>
</tbody>
</table>

**Note:** You cannot change the data type if another question depends on this question.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomize answers</td>
<td>Check box for displaying the possible answers for this question in a random order whenever the question appears. Answer preference is sometimes influenced by the order in which answer options appear, which can result in biased results. Randomizing answer options can help prevent this bias. This check box is available only if the Data type is Choice or Likert Scale.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Randomizing answer options for certain questions may make those questions confusing for users. In general, only randomize answer options that do not follow a logical order.</td>
</tr>
<tr>
<td>Scale definition</td>
<td>Setting that determines whether lesser or greater numerical values equate to a good score in quiz result calculations. Select Low if lesser numerical values are better. Select High if greater numerical values are better. The default value is High. This field is available and required unless the Data type is Date, Date/Time, or String. The results for these data types are not included in results calculations. When the Scored check box is selected, the scale value is set to High and the field is hidden.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>For information about how to set the scale definition for data types that do not require you to set a numerical value, see Data types for assessments on page 317.</td>
</tr>
<tr>
<td>Min</td>
<td>Lowest numerical value that can be used as an answer option. This field is available and required only if Data type is Number, Duration, or Percentage.</td>
</tr>
<tr>
<td>Max</td>
<td>Highest numerical value that can be used as an answer option or scaled value. This field is available and required if Data type is Number, Duration, or Percentage.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>String option</td>
<td>Setting for the appearance of a string field in a question. This field is available when the question type is String. The string options are:</td>
</tr>
<tr>
<td></td>
<td>• Single line: Single line text field 40 characters in length that allows strings of any length.</td>
</tr>
<tr>
<td></td>
<td>• String line wide: Full page width text field that allows a single line entry of any length.</td>
</tr>
<tr>
<td></td>
<td>• Multiline: Full page width multiline text field that allows word wrap and returns.</td>
</tr>
<tr>
<td>Scored</td>
<td>Check box for using answers in scoring the category results and showing them in the quiz results. Scored questions are available for these data types:</td>
</tr>
<tr>
<td></td>
<td>• Choice</td>
</tr>
<tr>
<td></td>
<td>• Likert Scale</td>
</tr>
<tr>
<td></td>
<td>• Template</td>
</tr>
<tr>
<td></td>
<td>• Checkbox</td>
</tr>
<tr>
<td></td>
<td>• Yes/No</td>
</tr>
<tr>
<td></td>
<td>Selecting this check box hides the Scale definition field and sets the value in that field to High.</td>
</tr>
<tr>
<td>Correct answer</td>
<td>[Required] Desired answer to a scored question. This field is available when the Scored check box is selected.</td>
</tr>
<tr>
<td>Template</td>
<td>Answer template to use with this question. This field is available when the Data type is Template.</td>
</tr>
<tr>
<td>Related List</td>
<td></td>
</tr>
<tr>
<td>Assessment</td>
<td></td>
</tr>
<tr>
<td>Metric Definitions</td>
<td>List of options for this question. This related list is available only if the Data type is Choice or Likert Scale.</td>
</tr>
</tbody>
</table>

Create answers for questions

Questions with Choice or Likert Scale data types must have defined answer options, called metric definitions.

When you create a question with one of these data types, the Assessment Metric Definitions related list appears. Each metric definition appears as one answer option for a question on a quiz. For example, the question What type of men’s trousers are not permitted by company policy? might have these three answers configured as metric definitions: Slacks, Jeans, and Casual cotton.

To create an answer option, open the question record you want to edit and click New in the Assessment Metric Definitions related list. Be sure to give each option a Value. The system uses this number to establish the order in which the answers appear in the quiz.
Distribute a quiz

When you finish configuring the answers for the quiz questions, you are ready to distribute the quiz.

You can send the quiz to all the category users configured for the quiz or to a single category user.

1. Navigate to Quiz Management > Quizzes.
2. Open the quiz record, and click Publish. The quiz is placed in the Published state, and it is sent to all its category users. You can edit and resend published quizzes. See Modifying Published Quizzes to learn how various modifications affect the quiz contents.
3. To resend a quiz, click the appropriate button:
   - Assign Quiz: Send the quiz to one category user.
   - Send Quizzes: Send the quiz to all of its category users.

Note: These buttons are hidden if there are no category users defined for the quiz.
Modify a published quiz

You can edit a quiz even after it has been distributed, with these results:
• Added questions are available only on quizzes that are distributed after this change.
• Changes to existing questions are immediately available to users before the quiz is submitted or during the retake period. This includes changes to the answers, such as additional choices or changes to the data type.
• Deleted questions are also deleted from the distributed quizzes in users' queues.

Edit a quiz
You can update a quiz after the quiz has been distributed.
Role required: assessment_admin or admin
• Questions that you add are available only on quizzes that are distributed after the update.
• Before a quiz is submitted or during the retake period:
  • Changes to existing questions are immediately available to users. This includes changes to the answers, such as additional choices or changes to the data type.
  • Deleted questions are deleted from distributed quizzes in users' queues.

Open the quiz: Group the Service Details form with one of the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes &gt; Quiz Designer</td>
<td>Point to the menu icon in the quiz header bar and select Load Quiz.</td>
</tr>
<tr>
<td>Quizzes &gt; Quizzes</td>
<td>Right-click a quiz in the list and select Quiz Designer.</td>
</tr>
<tr>
<td>Quizzes &gt; Quizzes</td>
<td>Open a quiz from the list and click Quiz Designer in the header of the Assessment Metric Type form.</td>
</tr>
</tbody>
</table>

Configure a scored question
Only scored questions are considered when calculating category and quiz results.
Role required: assessment_admin or admin
A question must have a correct answer specified to be scored. Only results for scored questions are displayed in the quiz scorecard.
1. Navigate to Quizzes > Quizzes.
2. Open the quiz containing the questions you want to mark as scored.
3. In the Metric Categories related list on the Assessment Metric Type form, select the category for the questions you want to mark as scored.
4. In the Assessment Metrics related list on the Metric Category form, select a question from the list.
5. In the Question Type section of the Assessment Metric form, select the Scored check box. This check box is not available if the question's data type is not supported for scoring.
6. Select the Correct answer for the question.
7. Click Update.
8. Repeat the process for all the questions in the category that you want the system to score.
9. Return to the list of metric categories, select another category, and configure scoring for the appropriate questions in that category.

Configure a template question
You can configure template questions when designing quizzes.

Role required: assessment_admin or admin

1. Drag the Template data type icon into a category container.
2. Click the gear icon in the question title bar to open the template properties dialog box.
3. Select a predefined scale from the list.

Question entry fields appear for that template.

4. Enter one or more questions that are appropriate for the template.
5. Click the arrow to the right of a question to configure its properties. You must provide a name for each question.
6. Click the up or down arrow to move between questions, or click the back arrow to return to the template properties dialog box.
7. Configure the properties for the remaining questions.
8. Click the X icon to close the template properties dialog box and save your settings.

Enable a quiz retake
You can configure a quiz to allow recipients to resubmit their answers as many times as they like, until the quiz's due date.

Role required: assessment_admin or admin

Results are not calculated until the quiz’s configured duration has elapsed. The card in the user's queue remains visible until the quiz's due date and displays a button to allow retakes.

1. Navigate to Assessments > Metric Definition > Types.
2. Remove the Evaluation method = Assessment filter condition so you can see all the records in the list.
3. Open the quiz.
4. In the Assessment Metric Type form, select the Allow retake check box and save the record.
View a quiz result
You can view quiz results for each question and category, or view the quiz scorecard for a detailed breakdown.

Role required: assessment_admin or admin

Quiz results are stored in the Metric Result [asmt_metric_result] table and display recipients' answers to each question in a category.

Navigate to **Quizzes > Quiz Results**.

---

Quiz reports
Quizzes provide several global reports so that assessment administrators can view important statistics.

You can share these reports with specific users or groups and change the display options.

For detailed field information and reporting options, click the link for the chart **Type**. To sort a column in ascending or descending order, click the arrow in the column heading.

**Table 100: Quiz reports**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes by State</td>
<td>State of distributed quizzes. This chart displays the percentage of distributed quizzes that are ready to take, in progress, complete, or canceled.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Pie Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Assessment Instance [asmt_assessment_instance]</td>
</tr>
<tr>
<td>Total Questions by Quiz</td>
<td>Total number of questions for all categories in each quiz.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Questions by Data Type</td>
<td>Total number of questions in all quizzes by data type.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td>Correct Answers by Assigned User</td>
<td>Total number of scored questions answered correctly by each assigned user.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Number of Correct Answers</td>
<td>Total number of correct answers for each scored question.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Incorrect Answers by Assigned User</td>
<td>Total number of scored questions answered incorrectly by each assigned user.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Number of Incorrect Answers</td>
<td>Total number of incorrect answers for each scored question.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
</tbody>
</table>

View a quiz scorecard
You can view scorecards for a quiz record.
Role required: assessment_admin or admin
For detailed information about scorecards, see Quiz scorecards.

1. Navigate to **Quiz**es > **Qui**zzes.
2. Open a quiz record.
3. Under **Related** Links, select **View** Scorecard.

   The scorecard link is hidden if there are no quiz results to report.

4. The scorecard contains a header that displays the name of the quiz and a section that displays results or a comparison of ratings. You can select from these principal views:
   • Category Results
   • Question Results
   • Average Ratings
   • History

Quiz scorecards
The Quizzes application prepares printable scorecards.
A scorecard analyzes category and question responses and compares current ratings with previous ratings. You can examine ratings over time, compare question ratings, or compare the ratings of all categories. All ratings are averages for the time range selected.

The system dynamically updates a scorecard each time you view it, so the ratings reflect recently completed quizzes.

Category results
The Category Results view is a stacked bar chart of responses to all questions in a category.

Select the category to display from the choice list above the chart. Category results are only calculated for scored questions

This view displays responses that use the following data types:

- Checkbox
- Choice
- Likert Scale
- Number
- Template
- Yes/No

**Note:** The Checkbox and Yes/No data types are combined into the Boolean data type in the Quiz designer.
Figure 136: Quiz category results
To view details about a specific response to a question, point to the colored segment representing a specific response. The chart displays the count for those responses and the percentage it represents of the total responses to that question.

Figure 137: Quiz category result details

Question results
The Question Results view shows the results for all questions in a quiz.
Select a question by name from the choice list to display the results in a pie chart or a bar chart, based on the data type.

Pie chart

The pie chart shows question results for these data types:
- Checkbox
- Choice
- Likert Scale
- Number
- Template
- Yes/No

Note: The Checkbox and Yes/No data types are combined into the Boolean data type in the Quiz designer.
Company Policies

Quiz Scorecard

Question Results | Office Supplies

Office Supplies

- (empty) = 3 (27.27%)
- Batteries = 2 (18.18%)
- Employees cannot take office supplies for personal use = 3 (27.27%)
- Printer paper = 2 (18.18%)
- Staplers, if they bring them right back = 1 (8.96%)

Figure 138: Quiz scorecard question results - pie chart
Bar chart

A bar chart appears when question results use this data type:

- Percentage

By default, all results for percentage questions use a report range of 20% segments. To configure a report range, navigate to Reports > Administration > Report Ranges.

Average ratings

The Average Ratings view displays a bar chart of the weighted average rating for each question in a category.

Use this view to learn how individual questions affect the overall rating for the category. Select a category from the second choice list above the chart. Ratings are only calculated for scored questions.
Figure 139: Average Ratings view

To view the effect of each question's ratings on the entire category's ratings, point to the colored bar. The pop-up box shows the percentage of the total ratings represented by each individual question's weighted average.
Scorecard history

The History view compares the current ratings for the categories and their questions with ratings from the previous three years or four quarters.

Ratings that have declined are highlighted in red and display negative numbers. Ratings that have improved are highlighted in green with positive numbers. Arrow icons beside the values in the Diff column indicate the trend of the current ratings against the previous ratings. Ratings are only calculated for scored questions questions.

Point to a category to display a line chart that shows the rating trend for that category. Click a category to view the Metric Category form containing the questions.

• **3 Years**: To calculate the current ratings, the system averages the ratings from the trailing twelve month (TTM) period. The Diff column shows the discrepancy between the current ratings and the previous calendar year's ratings.
Figure 142: Scorecard history - 3 years

- **4 Quarters**: Quarterly quizzes compare the average rating for each question and category in the current quarter against the average ratings from the previous four quarters. The **Diff** column shows the discrepancy between the current ratings and the previous quarter's ratings. The column labels count backward, by quarter from the current quarter. For example, if the current quarter is the 3rd quarter of 2013, then the previous quarters appear as 2nd [2013], 1st [2013], 4th [2012], and 3rd [2012]. All four of the previous quarters appear, whether or not there was any data for those quarters.
Category results contain values that represent an evaluated record's performance in a specific metric
category. Each category result record stores data from one assessment group. Category result values are
calculated based on category and metric weights and from metric result values for the same category.

Role required: assessment_admin or admin

Note: The system does not include metric results from certain responses in category result
calculations. To compensate, the system adjusts the weight of the other normalized metric result
values within the same category.

1. Navigate to Assessments > Results > Category Results.
2. Click the reference icon ( ) next to an assessment group number to open the category result record.
3. View the Assessment Category Result form.
   All fields on the form are read-only.

   **Table 101: Assessment Category Result form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Name of the metric category that the category result values apply to.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Rating | Rating is used on scorecards and decision matrixes. Rating is updated dynamically as users complete assessments. Rating is calculated as:  

\[
\text{Rating} = \frac{\text{Sum of all normalized values that share a particular assessment group, category, and assessable record}}{\text{Number of assessment instances for the assessment group}}
\]

For example: In the Assessment Category Results table example, the assessment group ASG0000801 with category Politeness from group NY DB (the assessable record) has a Rating value 8.86.  

On the associated Metric Results table, you can see that there are 4 instances of this group: AINST0000801, AINST0000802, AINST0000803, and AINST0000804.  

The system calculated the rating using  

\[
(5.14 + 5.14 + 1.43 + 5.71 + 5.14 + 4.29 + 4.29 + 4.29) / 4 = 8.86
\]
Create a decision matrix

The name and labels on a decision matrix are closely linked to the categories you choose for the axes. For that reason, first select the metric type and define the X and Y axes before you fill in other fields on the Decision Matrix form.

Role required: assessment_admin or admin

1. Navigate to Assessments > Admin > Decision Matrixes.
2. Click New and then select the metric Type to plot results for.
3. Right-click the form header and select Save.
   The X Axis and Y Axis related lists appear.
4. Define which metric categories the axes represent. Each axis can represent a single category or multiple categories. You must specify at least one category per axis in order for the decision matrix to render properly.
The available categories are limited to those associated to the metric type selected. To avoid confusion, always update the text for the axis and quadrant labels after editing or adding metric categories to an axis.

Figure 144: Decision matrix axes

5. Fill in the remaining fields on the Decision Matrix form (see table) and save the record.
### Decision Matrix

<table>
<thead>
<tr>
<th>Quadrant Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadrant label color:</td>
</tr>
<tr>
<td>X-Axis label:</td>
</tr>
<tr>
<td>Plotted item color:</td>
</tr>
<tr>
<td>Top left color:</td>
</tr>
<tr>
<td>Bottom left color:</td>
</tr>
<tr>
<td>Y-Axis label:</td>
</tr>
<tr>
<td>Highlight item color:</td>
</tr>
<tr>
<td>Top right label:</td>
</tr>
<tr>
<td>Top right color:</td>
</tr>
<tr>
<td>Bottom right label:</td>
</tr>
<tr>
<td>Bottom right color:</td>
</tr>
</tbody>
</table>

### Related Links

**View Matrix**

**X Axis**

- [Decision matrix = Importance vs User Satisfaction](#)

**Y Axis**

- [Decision matrix = Importance vs User Satisfaction](#)
**Note:** For color fields, either HTML color names or hexadecimal (hex) values are acceptable. For hex values, the # character is optional. Values are not case-sensitive. For example, all of the following values are valid: LightGray, lightgray, #D3D3D3, d3d3d3.

---

### Table 102: Decision Matrix form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Name of the decision matrix, which appears as the title on the decision matrix page. It is recommended to include the axis categories in the name. For example, if you selected <strong>Importance Rating</strong> as the X-axis and <strong>User Satisfaction</strong> as the Y-axis, you might name the matrix <strong>Importance vs. User Satisfaction</strong>.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Metric type associated with this decision matrix. Only results for assessable records of the selected metric type are plotted on the decision matrix.</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>Determines whether or not this is the default decision matrix. The default decision matrix opens when you click <strong>View Matrix</strong> on a scorecard. Select the check box to set the matrix as the default decision matrix. The system prevents you from creating more than one default decision matrix per metric type. If there is already a default decision matrix for the type and you try to save a different matrix with the check box selected, an error message appears.</td>
</tr>
<tr>
<td><strong>Quadrant Design Section</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Quadrant label color</strong></td>
<td>Color of the label text for the quadrants. Each quadrant label displays in the center of the quadrant. You can enter an HTML color name or hex value for this and the other color fields.</td>
</tr>
<tr>
<td><strong>X-Axis label</strong></td>
<td>Label text for the X-axis of the decision matrix. It is recommended to include the metric category name in the label text. For example, for an X-axis category of <strong>Importance Rating</strong>, the X-axis label is <strong>Importance</strong>.</td>
</tr>
<tr>
<td><strong>Plotted item color</strong></td>
<td>Color used to display plotted items. For more information, see <a href="#">View a Decision Matrix</a>.</td>
</tr>
<tr>
<td><strong>Top left label</strong></td>
<td>Label text for the top left quadrant. For an X-axis labeled <strong>Importance</strong> and Y-axis labeled <strong>Support</strong>, you might label the top left quadrant <strong>Low importance, high support</strong>.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top left color</td>
<td>Fill color for the top left quadrant.</td>
</tr>
<tr>
<td>Bottom left label</td>
<td>Label text for the bottom left quadrant.</td>
</tr>
<tr>
<td>Bottom left color</td>
<td>Fill color for the bottom left quadrant.</td>
</tr>
<tr>
<td>Bottom left label</td>
<td>Label text for the bottom left quadrant.</td>
</tr>
<tr>
<td>Bottom left color</td>
<td>Fill color for the bottom left quadrant.</td>
</tr>
<tr>
<td>Y-Axis label</td>
<td>Label text for the Y-axis of the decision matrix.</td>
</tr>
<tr>
<td>Highlight item color</td>
<td>Color of highlighted plotted items. When you view a decision matrix from an</td>
</tr>
<tr>
<td></td>
<td>assessable record's scorecard, the assessable record plotted item appears</td>
</tr>
<tr>
<td></td>
<td>in the highlight color. Specify a highlight color that is different than</td>
</tr>
<tr>
<td></td>
<td>the <strong>Plotted item color</strong>.</td>
</tr>
<tr>
<td>Top right label</td>
<td>Label text for the top right quadrant.</td>
</tr>
<tr>
<td>Top right color</td>
<td>Fill color for the top right quadrant.</td>
</tr>
<tr>
<td>Bottom right label</td>
<td>Label text for the bottom right quadrant.</td>
</tr>
<tr>
<td>Bottom right color</td>
<td>Fill color for the bottom right quadrant.</td>
</tr>
<tr>
<td>Related Lists</td>
<td></td>
</tr>
<tr>
<td>X Axis</td>
<td>Lists categories that define the X-axis of the decision matrix.</td>
</tr>
<tr>
<td>Y Axis</td>
<td>Lists categories that define the Y-axis of the decision matrix.</td>
</tr>
</tbody>
</table>

### Decision Matrixes
Assessment results obtained by questionnaires and scripted metrics can be mapped to decision matrixes. Assessment administrators can view and create these dynamically updated graphs, which make it possible to compare assessable records by category. Decision matrixes display data from a trailing twelve month (TTM) period.
The decision matrix page has these components:

Table 103: Decision matrix components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>Select the subset of assessable records you want to view. The filter options available vary by metric type, based on the Filter field and Filter condition field settings for each type.</td>
</tr>
<tr>
<td>Scale</td>
<td>Select the scale for the decision matrix. The greater the scale, the larger the decision matrix appears.</td>
</tr>
<tr>
<td>Decision matrix</td>
<td></td>
</tr>
<tr>
<td>X- and Y-axes</td>
<td>Each axis represents one or more metric categories. If multiple categories are used for an axis, their respective weights determine the positioning of the plotted items.</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Plotted items</td>
<td>The labeled points you see on a decision matrix, called plotted items, represent averages of <em>category result</em> data for assessable records. Point to a plotted item label to view a rating summary for that assessable record. Click a plotted item label to view the <em>scorecard</em> for the assessable record.</td>
</tr>
</tbody>
</table>

![Decision Matrix Diagram](image)
Plotted item rating summaries:

If you point to a plotted item label on a decision matrix, a rating summary appears. The summary displays the assessable record’s average ratings for each axis. If an axis represents one metric category, the ratings are calculated averages from results for that category. If an axis represents multiple categories, the ratings are calculated averages from weighted results for all of the categories.

The summary shows:

- Current rating
- Difference between the current rating and the rating from the previous year
- Ratings from each previous year, going back three years

Figure 145: Decision matrix rating summary

View a decision matrix

How to view decision matrix.

Role required: assessment_admin or admin

1. Navigate to Assessments > Admin > Decision Matrixes.
2. Click a Name to open the decision matrix record.
3. Under Related Links, click View Matrix to view the decision matrix page.
Take an assessment

Assessments that are assigned to you appear in your assessment and survey queue as a card that displays the assessment name, state, due date, and the associated incident number. Click **Take Survey** when you are ready.

Role required: none

Overdue surveys and assessments are marked with a red icon and red due date. You must answer every required question, indicated by a red bar, before you can submit the assessment as complete. If you start to take an assessment but cannot complete it, save your responses and return to it later. When you have answered all the questions and are satisfied with the responses, submit the assessment.

By default, you cannot modify your answers to an assessment after submission. However, if the administrator has configured an assessment to allow retakes, you can edit your answers and resubmit the questionnaire. Completed assessments configured for retake remain in the queue until their due date and display the **Modify Assessment** button on the card.

1. Navigate to **Self-Service > My Assessments & Surveys**.

   **Note:** Users with the assessment_admin role can display other users' assessments and surveys in addition to their own. Use the **Show all** and **Show assigned to me** related links to show and hide assessments and surveys. Click a card assigned to another user to open the associated metric type or survey definition.

2. On the assessment card, note the due date.

   **Note:** Depending on the configuration, you may receive email notifications to remind you of the due date.

3. Click **Take Assessment** to open the questionnaire.

   Assessment questionnaires are arranged in sections: first by record, then by category. Each record appears as a section title. Questions for the record appear below, grouped by category. Each category appears as a subsection below the record name. Click the collapse icon (−) or expand icon (+) to hide or show the questions in a category, or all the categories and questions for a record. Colored bars indicate the status of each question.
4. Answer each question to the best of your ability. Point to a question for more information.

If you are unsure of how to respond to a question or if a question does not apply to a particular record, select Not Applicable, if available.

5. Read any assertions present at the end of the assessment and acknowledge with a signature, if required.

A signature can require you to select a check box or authenticate your full name, which the system displays in a read-only field. You cannot submit your answers to the assessment until you provide the required signature.
6. Save or submit the assessment.
   - **Save**: Saves your responses without submitting them. You can close the questionnaire and access it later from your queue.
   - **Submit**: Submits the completed assessment when you are finished.

7. If prompted, enter your user name and password to verify your full name signature.
   If all the questions are answered with valid values, a success message appears. If the system detects an unanswered mandatory question or invalid response, the assessment is not submitted, and a message appears at the top of the questionnaire explaining the error. Questions with problems are temporarily highlighted.

8. You can modify your responses to the assessment until its due date. To update your answers and resubmit an assessment that permits retakes, click **Modify Assessment**.

*Assessment questionnaires*
In the assessment process, users complete assessment questionnaires on topics of interest to provide subjective data for future business decisions.
No special role is required to complete an assessment. You are eligible to complete assessments that are assigned to you. For example, you might be expected to evaluate vendors your organization works with, based on traits related to customer service. The system stores your responses so decision makers can compare the performance of the items you evaluate. You and your manager may receive email notifications as reminders of important assessment information.

Metric types and assessable records

In the Assessments application, assessment administrators create and administer metric types and assessable records.

A metric type defines a set of records an organization wants to evaluate, such as vendors, projects, or employees. For each type, the system generates unique assessable records that link the type to records that need to be evaluated, such as the individual records for the vendors Amazon and Intel. There may be multiple assessable records for the same source record if the source record meets the criteria for more than one type. For example, you might want to evaluate a record on the Company table, such as Intel, as a vendor and as a manufacturer, with different categories and metrics.

For configuration suggestions, see Administrator tasks on page 312.

Assessable records

An assessable record links a source record you want to evaluate, such as the company record for Amazon or the user record for a sales representative, to a metric type, such as vendors or employees.

You use assessments to evaluate the assessable record. The system generates assessable records from the source records that match the table and conditions set on the Assessment Metric Type form. You evaluate the assessable records with metric categories and metrics, which define traits and values to assess. For metric types with the On demand schedule type, you can generate on-demand assessments from the Assessable Record form. This method of assessment generation makes it easy to create and preview short questionnaires or to quickly obtain assessment results for specific assessable records.

Delete an assessable record

When you delete an assessable record, the system deletes any stakeholders for the record.

Role required: assessment_admin or admin

Note: If a source record is deleted, the system deletes the associated assessable record. To delete the source record, you must first delete all associated metric results and category results.

Delete the assessable record:

• To delete a single record, open the record and click Delete.
• To delete multiple records, use the Assessable Records list.

View an assessable record

View the Assessable Record form to edit preferences and perform various actions.

Role required: assessment_admin or admin

1. Navigate to Assessments > Assessable Records.
2. Open a record from the list.
   By default, the list displays only assessable records with Active metric types.
3. On the Assessable Record form, edit fields and perform other actions as necessary (see table).
# Table 104: Viewing Assessable Records

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Read-Only] Assessable record name based on the <em>display value</em> of the source record. The name appears on assessment questionnaires.</td>
</tr>
<tr>
<td>Source</td>
<td>[Read-Only] Source record the assessable record is linked to. The <strong>Source</strong> reflects the table name and source record display value. For example, if the <strong>Name</strong> field is the display value for the Company table, the assessable record for a company record named <strong>Amazon</strong> has the <strong>Source</strong> value <strong>Company: Amazon</strong>.</td>
</tr>
<tr>
<td>Live feed</td>
<td>Check box that, when selected, creates a live feed group for the assessable record, which appears on the scorecard. If you clear the check box after a live feed group has been created, the system deletes the live feed group and all its messages.</td>
</tr>
<tr>
<td>Type</td>
<td>[Read-Only] Metric type from which the assessable record was generated.</td>
</tr>
<tr>
<td>Decision matrix</td>
<td>Check box that, when selected, enables this assessable record’s results data to appear on decision matrixes of the same metric type. Decision matrixes are graphs that plot the assessment results for multiple assessable records. If you clear the check box, the assessable record still appears on the default decision matrixes if you click the <strong>View Matrix</strong> related link on the assessable record's scorecard.</td>
</tr>
<tr>
<td>Live feed group</td>
<td>[Read-Only] Live feed group for this assessable record. When you select the <strong>Live feed</strong> check box and save the record, the system populates this field.</td>
</tr>
</tbody>
</table>

**Related Links**

- **View Scorecard**
  
  Opens the scorecard for the assessable record.
### Related Lists

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>All metric categories associated with the assessable record. An assessable record must be associated to a category to be evaluated. Click <strong>Edit</strong> to add or remove category associations. Note that it is often more efficient to associate assessable records to categories using the Metric Category form.</td>
</tr>
<tr>
<td>Category users</td>
<td>All stakeholders who can take assessment questionnaires about this assessable record. Click <strong>Edit</strong> to create and delete stakeholders. For more information about this and other methods of creating stakeholders, see <strong>Stakeholders</strong>. This related list is available only when the associated metric type has the <strong>Scheduled</strong> schedule type.</td>
</tr>
</tbody>
</table>

### Approvals

Approvals can be defined for all tasks and allow users or groups to be associated with a task, for the purpose of either approving or rejecting that task. Approvals are defined by navigating to **System Policy > Approvals**.

The following information defines an approval:

**Table 105: Approval definition information**

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approver</td>
<td>A reference to the user who is responsible for approving the related record.</td>
</tr>
<tr>
<td>Approving</td>
<td>Choices are:</td>
</tr>
<tr>
<td></td>
<td>• Not Yet Requested (This state indicates that you are not yet asking your approvers to approve this request. Until you set the status to <strong>Requested</strong> they will receive no email notifications about the request.)</td>
</tr>
<tr>
<td></td>
<td>• Requested</td>
</tr>
<tr>
<td></td>
<td>• Approved</td>
</tr>
<tr>
<td></td>
<td>• Rejected</td>
</tr>
<tr>
<td>Comments</td>
<td>A journal field for storing comments regarding the approval.</td>
</tr>
</tbody>
</table>
Approval engines

The differences in the way that companies handle their approvals, as well as the differences between approvals for the various applications (such as Service Catalog Requests and Change Management), calls for supporting flexibility in setting up approvals within applications. This flexibility is provided through the selection of an "approval engine" that is used to manage the approvals for each of the Task tables (that is, all tables that extend the Task table).

There are three different approval engine options available for each Task table.

Table 106: Approval engine options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval Rules</td>
<td>A simple set of rules that are evaluated until one matches for the Task table. The matching approval rule is used to create the users that are to approve the task. Set up approval rules by navigating to System Policy &gt; Approvals.</td>
</tr>
<tr>
<td>Process Guides</td>
<td>A sequence of approval steps over which you may control how approvals and rejections are handled. This option is deprecated and should not be used.</td>
</tr>
<tr>
<td>Turn off Engines</td>
<td>Turn off both approval engines for this Task table. This option should be selected and is made read-only when a workflow is used to manage the approval process for the table.</td>
</tr>
</tbody>
</table>

**Caution:** Not turning off the approval engines might have a performance or behavioral impact on your instance.

Set up an approval engine

1. Navigate to System Properties > Approval Engines.

The following page appears with the Approval Engine option for each Task table in the system. If the Approval Engine option is greyed out and shows Turn engines off, read the Notes in the same row. The most common reason an approval engine is turned off is that a workflow is managing the approvals on the table. Having the approval engine turned off prevents conflicts with the workflow that could cause a range of issues. If you want to use an approval engine on the table, set the workflow to inactive.
2. Select the *approval engine option* for each Task table from the choice list.
3. Click *Save*.
   These preferences are saved as system properties that are named glide.approval_engine.<table_name>.
Approval rules

Many organizations rely on an approval process to ensure that requests are reasonable and fit an organization's budget.

The service catalog can use these classes of approvals:

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gating approvals</td>
<td>Must occur before a request can be initiated. For example, allow a manager to reject an employee's request for a company car.</td>
</tr>
<tr>
<td>Process approvals</td>
<td>Take place within an execution plan process that has been initiated. For example, allow the security group to reject a request for access to SSN even though the employee's manager approved it.</td>
</tr>
</tbody>
</table>

Note: To enable approval processes to operate smoothly, make sure that the appropriate users have the correct role, and that the role grants access to the necessary tables for users in all the relevant departments and domains.

Set automatic approval rules

Approval rules can automatically set the approval state to something other than Not yet requested. As a result, an approval rule can create a set of approvers. You can also start the approval process by setting the approval state to Requested.

Prerequisites

Role required: admin

Approval rules have two new fields:

- **Run rule before**: If true, the approval rule runs before the record is inserted/updated.
- **Set State**: If this rule applies, then the task record's approval state is automatically set to this value.

Note: The Set State field only behaves as expected if the Run rule before check box is enabled.

- In the example below, this rule automatically sets the state of the task to Approved thereby auto-approving the task.
Gating approvals

A gating approval acts as a gate through which a request must pass before it can start.

Until all gating approvals are met, no notifications go out, no tasks get sent to technicians, and nobody starts working on the request in question.

Generate gating approvals with:

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval rules</td>
<td>Can apply to the service catalog as well as any other task table.</td>
</tr>
<tr>
<td>Item-based approvals</td>
<td>Flag specific catalog items as requiring specific approvals. Any requests for these items automatically require these approvals.</td>
</tr>
</tbody>
</table>

Set up a gating approval via an approval rule

You can set up a gating approval via an approval rule.

Role required: admin

1. From the left navigation pane, select **System Policy > Approval Rules**.
2. Click **New**.
Table 109: Approval rules

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of this rule.</td>
</tr>
<tr>
<td>Table</td>
<td>Task table to which this rule applies. For most service catalog approvals, select Request.</td>
</tr>
<tr>
<td>Note:</td>
<td>The list shows only tables and database views that are in the same scope as the approval rule.</td>
</tr>
<tr>
<td>Active</td>
<td>Indicator of whether the rule is active (defaults to true).</td>
</tr>
<tr>
<td>Run Rule Before</td>
<td>Indicator of whether the rule runs before or after the request record is saved. For most approvals, select this check box.</td>
</tr>
<tr>
<td>User</td>
<td>User who must approve this request (can be empty).</td>
</tr>
<tr>
<td>Group</td>
<td>Group that must approve this request (can be empty).</td>
</tr>
<tr>
<td>Set State</td>
<td>Value of the approval field on the task in after this rule runs. In most cases, select Requested.</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition under which the rule applies.</td>
</tr>
<tr>
<td>Script</td>
<td>An optional server script to programmatically specify who the approver should be. For example, for the one-line script current.requested_for.manager, ServiceNow checks the requested_for reference field on the current record. It then locates the manager field on the referenced record and assigns that person as the approver. For other examples, see the Script field on approval rules provided by ServiceNow.</td>
</tr>
</tbody>
</table>

Notes and limitations:

1. You can have as many rules as you want on a given table. If more than one rule applies, you'll get more than one approver.

2. You can't get duplicate approvers, for example, if two rules both want Fred Luddy to approve a particular request, the system will only create one approval entry for him.

3. By default all requests start out in a Not yet requested approval state. Approval notifications will not go out until the request's approval state is set to Requested. You can do that manually, or you can do it in script, but the easiest way to do it is to use the Set State field to automatically set the request to Requested.
Set up a gating approval based on the item being ordered

In addition to adding approvals via approval rules, you can also add approvals based on what kind of item is being ordered.

We can, for example, specify that all Blackberrys need to be approved by David Loo. To do so, navigate to the item in question and scroll to the related list of required approvers. There are two lists:

- **Approved By Group**: A list of groups that have to approve requests for this item
- **Approved By**: A list of users who have to approve requests for this item

---

**Approved By Group**

<table>
<thead>
<tr>
<th>New</th>
<th>Edit...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval group</td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td></td>
</tr>
</tbody>
</table>

**Approved By**

<table>
<thead>
<tr>
<th>New</th>
<th>Edit...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approver</td>
<td></td>
</tr>
<tr>
<td>David Loo</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 147: Approve list**

In the example above, this request must be approved by all members of the Hardware group and by David Loo.

**Notes and limitations:**

1. As with approval rules, you are protected against duplicate entries. Thus if David Loo is a member of the hardware group, as well as being a standalone approver, he will only get one approval request.

2. Item-based approved work in addition to rather than instead of approval rules so you can (and probably will) use both.

**Process approvals**

Once a request has passed its gating approvals, any relevant execution plans are initiated.

Those plans, in turn, create a sequence of required tasks. You can add an approval step to an execution plan, which is configured to occur at the appropriate point.

From the left navigation pane, select **Service Catalog > Execution Plans**, and then select the plan to which to add an approval step. Then click the **New Approval** button.
Figure 148: Add approval

The Approval Task screen appears. Just like a regular Service Catalog execution task, an approval execution task has:

- **Name**: The name of this task
- **Order**: Sequence of this task within the plan
- **SLA**: SLA to which this task applies
- **Delivery Time**: Time allowed for the completion of this task

After you create the task, right click the title bar and select **Save**. Two related lists appear at the bottom of the screen:

- **Approved By Group**: A list of groups that must approve the request before this task is complete
- **Approved By**: A list of users who must approve the request before this task is complete
Figure 149: Approval task

In the example above, this security approval task must be approved by Fred Luddy.

Note: When an in-process approval is rejected, that particular line item is canceled as well, but the request itself isn't necessarily canceled. Thus if one ordered a blackberry and a laptop, and the blackberry was rejected, the laptop request would continue being processed.

Approve with a process guide
Process guides work similarly to approval rules in that their execution is controlled via a condition.

The default version of approval tasks allows you to specify that the approval in question be approved by:

1. One or more specific people
2. One or more groups of people

You can optionally use Process Guides instead of approval tasks. Process guides are more flexible in that they allow for:

1. "Any of" or "All of" approvals
2. **Sequenced approvals**

You can link a process guide to an execution task.

1. From the left navigation pane, select **System Policy > Process Guides**.
2. Create a new guide.
3. Set the table to **Catalog task**.
4. Fill in a condition under which this guide should attach.

**Example #1**: Apply to all "Capacity Review" tasks.

```
Add Condition
   -  +  State ▼ is ▼ Open ▼
   -  +  and Delivery task ▼ is ▼ Capacity Review
```

**Example #2**: Apply to all "Capacity Review" tasks where the requester is in Atlanta.

```
Add Condition
   -  +  State ▼ is ▼ Open ▼
   -  +  and Delivery task ▼ is ▼ Capacity Review
   -  +  and Request item.Request.Requested for Location ▼ is ▼ Atlanta
```

**Process guide tips and tricks**:

1. All catalog tasks are generated when a request is first submitted, but tasks which aren’t active yet have a state of “pending”. So if you do not want to send out approval requests until a task has actually started, add "state=open" as part of your condition.

2. There is a "Default" process guide in the system for catalog tasks with a sequence number of 10,000. It behaves exactly the same way the old, pre-process guide code did in regards to approvals. Approvals are based on the execution of task-related lists.

_Schematic of a hypothetical approval process_

In the diagram below of a hypothetical approval process, the gating approval is color coded blue and an in-process approval is orange.
Approval summarizer formatter

The approval summarizer formatter creates the summary at the bottom of an approval form. The approval summarizer displays different information depending on what is being approved, such as a change request or a service catalog request. Following are two examples.
**Change Request**

<table>
<thead>
<tr>
<th>Number</th>
<th>CHG0000001</th>
<th>Requested by</th>
<th>David Loo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affected CI</td>
<td>Sales Force Automation</td>
<td>Type</td>
<td>Normal</td>
</tr>
<tr>
<td>Planned start date</td>
<td>2016-07-27 16:00:00</td>
<td>Risk</td>
<td>High</td>
</tr>
<tr>
<td>Planned end date</td>
<td>2016-07-27 18:00:00</td>
<td>Impact</td>
<td>3 - Low</td>
</tr>
<tr>
<td>Short description</td>
<td>Rollback Oracle Version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Performance of the Siebel SFA software has been severely degraded since the upgrade performed this weekend. We moved to an unsupported Oracle DB version. Need to rollback the Oracle Instance to a supported version.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The **Reject** button allows the approver to reject one or more requested items in a multi-item request, before approving the overall request. If a requested item is rejected, the workflow for that item never starts. The approver can then choose to **Approve** the item.

**Note:** When the overall request is approved, you must ensure this **Approve** button is hidden. If this button is used after request approval, the requested item workflow is canceled, leaving the stage in an inconsistent state.

Similarly, the **Approve** button on requested items should only appear before the overall request is approved or rejected.

---

**Summarizers**
Approval summarizers are stored in the Macro [sys_ui_macro] table.

From the left navigation pane, select **System UI > UI Macros**. Summarizers use a naming convention of approval_summarizer_+ '<table_name>' (for example, approval_summarizer_change_request is the summarizer for change requests, while approval_summarizer_sc_request is the summarizer for service catalog requests).

Each summarizer is written in Jelly script, which is used to define internal forms. The script is stored in the large XML field at the bottom of the UI Macro form.

**Change a summarizer**
You can modify existing approval summaries to include additional information.

These are advanced customizations that might not be appropriate for all implementations, and require creating a custom form.

1. Navigate to **System UI > UI Macros**.
2. Open the summarizer you want to change.
3. Copy the script to another location before editing, in case you need to revert it.
4. Modify the script.
5. Click Update.

Create a new custom summarizer
After you add a new table that has approvals to an instance, you can add a custom summarizer by creating a new UI macro.
1. Navigate to System UI > UI Macros
2. Click New.
3. Give the macro a name that follows the summarizer naming convention:
   approval_summarizer_<tablename>
4. Complete the rest of the form and click Submit.
5. Add the custom summarizer to the appropriate form.

Approval with e-signature
Approval with e-signature allows users to approve requests by re-entering their login credentials.
Approval with e-signature supports the following authentication credentials:

- User name and password matching a user in the local database.
- User name and password matching a user authorized by an external identity provider as part of a SAML 2.0 single sign-on integration.

Activate approval with e-signature
To activate approval with e-signatures, refer to the following topic.

Activate a plugin.

De-activate e-signatures
Use this procedure to de-activate e-signatures.
Although plugins cannot be removed, e-signatures can be disabled.
1. Navigate to System Definition > E-Signature Registry.
2. Set Enabled to False on any tables where e-signatures are no longer required.

Use e-signature approvals
Users can approve requests with an e-signature.

Users can approve requests with an e-signature by:
- Selecting the Approve or Reject option on a list context menu.
- Clicking the Approve or Reject button on a form.
- Changing a request record's State to Approved in either the list-editor or form.

Selecting any of these options presents the user with an Approver Authentication window, which requests the user's credentials again.

E-signature SAML properties
Configurable properties for e-signature with SAML 2.0 update 1.

**Note:** You DO NOT need to configure these properties if you activate Multiple Provider Single Sign-on.
### Table 110: E-Signature SAML Properties Form

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthnRequest URL for eSignature Authentication</td>
<td>Enter the URL that points to the SAML 2.0 Identity Provider AuthnRequest Consumer for eSignature Authentication. In most cases, this will be the same as the AuthnRequest URL used in general authentication. Leave this setting blank if you intend to use the same AuthnRequest Consumer URL that is used for general SAML 2.0 authentication in your instance.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The SAML 2.0 Assertion Consumer URL for eSignature authentication</td>
<td>In most cases, this URL will be: <a href="https://YOURINSTANCE.service-now.com/consumer.do">https://YOURINSTANCE.service-now.com/consumer.do</a>. However, if you employ a customized method of handling the SAML authentication for eSignature, you can set up your own consumer URL. If you are using Multi-Provider SSO, you do not need to use this property. Configure the Identity Provider form, add the <strong>Assertion Consumer URL for eSignature authentication</strong>, field, and set the URL in that field.</td>
</tr>
<tr>
<td>[com.snc.integration.saml_esig.approval_consumer_url]</td>
<td></td>
</tr>
<tr>
<td>The SAML 2.0 Assertion Consumer Index for eSignature authentication</td>
<td>If your Service Provider has more than one URL set for the AssertionConsumerURL, you can set the index to use for eSignature, starting with index 1 or more.</td>
</tr>
<tr>
<td>[com.snc.integration.saml_esig.assertion_consumer_service_index]</td>
<td></td>
</tr>
<tr>
<td>Authentication Pop-up Dialog Width</td>
<td>When a user approves a request using eSignature, a dialog allows the user to enter their credentials. This setting controls the width of that dialog box.</td>
</tr>
<tr>
<td>[com.snc.integration.saml_esig.popup_dlg_width]</td>
<td></td>
</tr>
<tr>
<td>Authentication Pop-up Dialog Height</td>
<td>When a user approves a request using eSignature, a dialog allows the user to enter their credentials. This setting controls the height of that dialog box.</td>
</tr>
<tr>
<td>[com.snc.integration.saml_esig.popup_dlg_height]</td>
<td></td>
</tr>
</tbody>
</table>

*Installed with approval with e-signature*

Installing approval with e-signature installs certain properties.

- Module - E-Signature Registry
- UI Action - Approve (on table sysapproval_approver, with no action name)
- UI Action - Approve (on table sysapproval_approver, with no action name)
- UI Action - Approve (on table sysapproval_approver, with the action name authenticated_list_approval)
- UI Page - form_login_validate_dialog
- UI Page - login_validate_dialog
- UI page: saml2_esignature_login, the re-authentication page that appears when an approver tries to approve a request.
- Properties: see [E-signature SAML properties](#)
- Client Script - Authenticate Approver
- Script Include - User
- Script Include - UserAuthentication
- Processor: eSigSaml2AssertionConsumer

Installing the plugin also disables the two out-of-the-box Approve UI Actions on the sysapproval_approver table.

*Select an approval table*

By default, activating the Approval with E-signature plugin enables e-signature for all tables for which a previous approval exists.

E-signature approvals can also be enabled on a table-by-table basis. To enable e-signatures for a table:

1. Navigate to System Definition > E-Signature Registry.
2. In **Table name**, use the drop-down list to select a specific table.
Table 111: Table Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>A table drop-down to select the table that requires e-signatures.</td>
</tr>
<tr>
<td>Enabled</td>
<td>If selected, e-signature is required. Clear this option to remove the e-signature requirement.</td>
</tr>
</tbody>
</table>

Set up an approval from a local database
To set up Approvals with e-signature using credentials from a local database:

1. Activate the Approval with E-Signature plugin.
2. Create user records for approval users.

Set up an approval from a SAML 2.0 authentication
Setting up e-signature with SAML 2.0 requires configuration on the identity provider and the instance.
The SAML Identity Provider (IdP) must support and honor the forceAuthn attribute in SAML Assertion Requests. E-signature will not function without this IdP setting.
To set up Approval with e-signature using credentials from a SAML 2.0 authentication:

1. Activate or upgrade to a SAML 2.0 Update 1 integration.
2. Activate the Approval with E-Signature plugin.
3. Enter the e-signature SAML properties.
4. Regenerate the service provider metadata and update it on the IdP.
5. Create user records for approval users.

Note: Customers on SAML 2.0 Update 1 no longer have to create custom UI pages for logging the user out or deleting session cookies.

Note: If you are a Life Science Customer using E-Signature, you must deactivate the User self-lockout prevention business rule. See KB0547061 for more information.

Set up e-signature approvals
The setup required depends on where user credentials are stored.

See the section that matches the source of your user credentials:
- Setting up approvals from a local database
- Setting up approvals from a SAML 2.0 authentication

Note: ADFS 2.0 does not support re-authentication requests that E-Signature requires.

Approval status
The approval status of a change request is determined by looking at the current status of all the approvers. If any approver has rejected the change, the approval status will be Rejected. If all approvers have approved the change, the approval status will be Approved. If all approvers are in the Not Requested
status or if there are no approvers, the change status will be Not Requested, otherwise the status will be Requested.

For added flexibility when creating approvals, including the ability to set up an "one of" approval where only one person of a group of approvers needs to approve, consider using Workflows.

Generate an approval using approval rules

The system can automatically generate an approval request to individuals or groups when specific criteria are met. The automatic generation of approval requests is driven using the System Policy feature.

In the sample below, a change opened in the category network is assigned to the System Administrator:

![Approval Rules](image)

**Figure 154: Approval Rules**

When an approver is automatically added based on approval rules, the status of the approval automatically defaults to "Requested".

Generate approvals using the approvers related list

It is possible to manually add approvers to a request.

Additional approvers can be added by clicking the Edit button in the Approvers section near the bottom of a request. When an approver is added manually, the status for that approver defaults to "Not Requested". When the status of the approver is changed to "Requested", the approver will be sent an email requesting approval action.

Generate approvals using workflows

Workflows are a powerful and flexible method of generating approvals. Use workflows to create group approvals and user approvals.

A variety of variables are available to fine-tune the approval process, including the actions that occur when approval or rejection take place. When a workflow activity generates an approval record, the system populates the **Workflow activity** field on the approval record with a reference to the activity. Do not use this field when creating business logic. For more information, see Approval and Rollback Activities.

**Note:** Conflicts can arise when the approval process for a table is managed by both the workflow engine and the approval engine. In general, if there is a workflow that manages the approval process for a table, the approval engine should be turned off for that table.

Multiple approvers

With multiple approvers, all approvers must authorize the request before the status will change to "Approved".

Should any approver reject the request, the status will immediately be set to "Rejected".
Receive notifications

Approval notifications will be sent at the following times:

- When an individual is assigned as an approver either automatically or manually. If a group is chosen, then all members of the group will be sent an email. By default, the email an approver receives will contain a "mailto" link that will allow the approver to either approve or reject the request directly from their email system.
- When the request reaches approved status, the person assigned to the request will receive an email indicating it has been approved.

The details contained in the emails and the points at which they are sent can be tailored using System Definition > Business Rules and System Policy.

Note for Blackberry users: In order to see the "mailto" links mentioned above to approve or reject a request (i.e. 'Click here to approve CHG55555' or 'Click here to reject CHG55555'), your Blackberry device must be using version 4.5 of their software which supports HTML emails. If your Blackberry device is using an earlier version, you will not be able to view or use the "mailto" links. However, as a workaround, users can reply to the email and add the statements state:approved or state:rejected within the body of the email before sending it to force the automatic approval/rejection functionality.

If you create an appropriate Inbound Email Action, you can let approvers respond to approval email notifications with a simple "yes" or "no" answer.

Dynamic approval forms

When you are looking at an approval request, the form has a context-appropriate summary of the item to be approved.

For example, if you're looking at a Change Management approval request, you'll see details from the relevant change request. For a Service Catalog approval request, you'll get details of the request.

![Approval Form Example](image)

Figure 155: Change request approval example
Execution order of scripts and engines

Scripts, assignment rules, escalations, and engines all take effect in relation to a database operation, such as insert or update. In many cases, the order of these events is important.

**Note:** Client-based code that executes in the browser, using Ajax or running as Javascript, will always execute before the form submission to the server.

The order of execution is as follows:

1. *Before* business rules: Scripts configured to execute before the database operation with an order less than 1000.

2. *Before* engines. The following are not executed in any specific order:
   - Approval engine (for task and sys_approval_approver tables)
   - Assignment rules engine (for task tables)
   - Data policy engine
   - Escalation engine
   - Field normalization engine
   - Role engine - keeps role changes in sync with sys_user_has_role table (for sys_user, sys_user_group, sys_user_grmember, and sys_user_role tables)
   - Execution plan engine (for task tables)
   - Update version engine - creates version entry when sys_update_xml entry is written (for sys_update_xml table)
   - Workflow engine (for default workflows)

3. *Before* business rules: Scripts configured to execute before the database operation with an order greater than or equal to 1000.

4. The database operation (insert, update, delete).
5. *After* business rules: Scripts configured to execute after the database operation with an order less than 1000.

6. *After* engines. The following are not executed in any specific order:
   - Label engine
   - Listener engine
   - Table notifications engine
   - Role engine - keeps role changes in sync with sys_user_has_role table (for sys_user, sys_user_group, sys_user_grmember and sys_user_role tables)
   - Text indexing engine
   - Update sync engine
   - Data lookup engine inserts or updates
   - Workflow engine (for deferred workflows)

7. Email notifications. The following are executed based on the weight of the notification record:
   - Notifications sent on an insert, update, or delete
   - Event-based notifications

8. *After* business rules. Scripts configured to execute after the database operation with an order greater than or equal to 1000.

Geolocation

The geolocation feature uses Google Maps to track users, plan efficient routes between locations, and assist in finding accurate travel times for applications that require this information, such as Work Management.

The system locates users from latitude and longitude information provided by their mobile devices or browsers. Tracking the location of certain users can improve business processes such as the completion of work order tasks. For example, in work management geolocation, the system updates an agent's position each time that agent updates a task record. When the system updates an agent's location frequently, agents and dispatchers can use features like auto-routing to create efficient schedules.

Geolocation features can be configured by users with the admin role.

Activate geolocation

An administrator can activate the Geolocation plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Installed with geolocation

The following components are installed with Geolocation:

Modified by Geolocation
The Geolocation plugin adds a field called *Geolocation tracked* to the User [sys_user] table. This field allows the system to track individual users by their geographical coordinates.

**Business rules installed with geolocation**

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoLocation Data updated</td>
<td>User [sys_user]</td>
<td>Populates the <strong>Latitude</strong> and <strong>Longitude</strong> fields on the Geolocation History [geo_history] table when the <strong>Latitude</strong> and <strong>Longitude</strong> fields are updated on the User [sys_user] table.</td>
</tr>
</tbody>
</table>

**Client scripts installed with geolocation**

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Geolocation on Task</td>
<td>Task [task]</td>
<td>Populates the <strong>Latitude</strong> and <strong>Longitude</strong> fields on a user record with GPS coordinates when that user loads any record on the Task table or a table that extends Task. This occurs only when geolocation tracking is enabled for the user.</td>
</tr>
</tbody>
</table>

**Properties installed with geolocation**

Geolocation provides the following configurable properties in Geolocation > Administration > Geolocation Properties.

The Google Maps plugin provides additional map properties that are used to implement and control specific geolocation features. You must have a private key for Google Maps API for Business to take advantage of the properties listed here.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.geolocation.allow.toll.roads      | Allow toll roads to be used. Allows the system to use toll roads when auto-routing an agent's tasks.  \  
  • **Type**: True/false    \  
  • **Default value**: True |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.geolocation.default.start.time</td>
<td>Default start time for all agents when no schedule is set. This value sets the start time for a day other than the current day, when no scheduled task exists or is continued from the previous day. This property uses a 24 hour clock.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: String</td>
</tr>
<tr>
<td>glide.geolocation.evening.rush.hours</td>
<td>Evening rush hour span, formatted as 14:30-16:00. All times are expressed as a range, separated by a dash with no spaces. Times not using this format are ignored. This property must be used with the work.management.morning.rush.hours and work.management.rush.travel.buffer properties and uses a 24 hour clock.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: String</td>
</tr>
<tr>
<td>glide.geolocation.history.cleanup</td>
<td>Number of days to keep history data. ServiceNow keeps agent geolocation history records for the number of days specified by this value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Integer</td>
</tr>
<tr>
<td>glide.geolocation.morning.rush.hours</td>
<td>Morning rush hour span, formatted as 06:30-08:00. This property uses a 24 hour clock. All times are expressed as a range, separated by a dash with no spaces. Times not using this format are ignored. This property must be used with the work.management.evening.rush.hours and work.management.rush.travel.buffer properties.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: String</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.geolocation.proximity               | Minimum distance an agent must move to be considered in a new location (in meters). This setting establishes a geolocation perimeter that prevents ServiceNow from creating unnecessary history records when an agent has multiple tasks in the same vicinity. An example of this might be when an agent has several tasks in one building. The system assumes that all task activity within the specified radius is part of a single record and does not create additional records when the agent moves from task to task around the building. | • **Type**: Integer  
  • **Default value**: 500                                                                                                                                                                                                                                                                                                                                                                                                          |
| glide.geolocation.proximity.location      | Maximum distance an agent can be from a specific location and still be placed at that location (in meters). This setting defines the maximum distance that an agent can be from a specified location and still be placed at that location by the system.                                                                                                                                                                                                                                                                                                | • **Type**: Integer  
  • **Default value**: 200                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| glide.geolocation.rush.travel.buffer      | Percentage to add to all rush hour travel times. The system uses this percentage to calculate schedules when auto-routing an agent. Use this property when both morning and evening rush hour times are defined. An example of a valid time buffer percentage is 15.                                                                                           | • **Type**: Integer  
  • **Default value**: 0                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| glide.geolocation.tracking.frequency      | Sets the length of the recurring interval, in seconds, that the system waits between attempts to update a user’s geographic coordinates. The system updates geographic coordinates only for users who have geolocation tracking enabled. Minimum amount of time between updating the user’s location (in seconds). This setting defines how long the system waits before updating an agent’s location, regardless of task activity or travel. This prevents ServiceNow from creating unnecessary history records if an agent in one location views a task multiple times. | • **Type**: Integer  
  • **Default value**: 300                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
### Table 115: Script Includes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeolocationAJAX</td>
<td>Utilities the system uses when making geolocation AJAX calls.</td>
</tr>
<tr>
<td>GeolocationUtils</td>
<td>Core utilities for geolocation functionality.</td>
</tr>
</tbody>
</table>

#### Geolocation Google key

Geolocation requires a Google Maps API for Business license to enable the use of Google driving time estimates and to use Google Maps. Applications, such as Work Management, have functionality that requires this license.

If you are using auto-routing or auto-dispatch in Work Management, both the Google Maps API for Business private key and the client ID must be set for precise time estimates. Without these values, driving times for auto-routing and auto-dispatch are not available, and a rough estimate or fixed time is used. Use these properties, found in **System Properties > Google Maps**, Google Maps, to enable the geolocation feature.

- **Client ID for Google Maps API for Business**: Google client ID for your organization. This ID is from Google and starts with `gme-`, such as `gme-mycompanyname`.
- **Private key for Google Maps API for Business**: Key required to get accurate driving time estimates from Google for some features of Geolocation. An example of an encoded key is: `vNIXE0xscrmjlyV-12Nj_BvUPaw="`

#### Geolocation history

The Geolocation History `[geo_history]` table contains the history of each tracked user's location over time.

To view geolocation history, navigate to **Geolocation > Geolocation History**. The record list shows time stamped location data for all users configured for geolocation tracking.

The following system properties, located in **Geolocation > Administration > Geolocation Properties**, control the data that appears in this table.

- **Minimum distance an agent must move to be considered in a new location (in meters)**: Perimeter that defines a single location in which multiple tasks are present. This property prevents the system from creating unnecessary history records when multiple tasks are in the same general location.
• Maximum distance an agent can be from a specific location and still be placed at that location (in meters): Distance that a user can be from a specified location and still be placed at that location when the system creates history data.
• Number of days to keep history data: Time period for which the system retains history data.
• Minimum amount of time between updating the user’s location (in seconds): Time period the system waits before updating a user's location, regardless of how often the user accesses tasks.

Location tracking

Geolocation adds the **Geolocation tracked** field to the User form. In the default view of a user record, select the **Geolocation tracked** check box to enable location tracking for the user.

When a user has geolocation tracking enabled, the system updates the user's geographic coordinates whenever the user opens, or loads, a record from the Task [task] table or from a table that extends Task. After this initial update, the system continues to update the user's geographic coordinates at a recurring interval if the user does not close or reload the record. To edit the length of this interval, navigate to Geolocation > Administration > Geolocation Properties and change the setting for the **property** that regulates the update interval for the user’s location. The default setting for this property is 300 seconds (5 minutes).

**Note:** The system can update geographic coordinates only for users who have location services enabled in their browsers. Even users who have the **Geolocation tracked** check box selected may be prompted by their browsers to share or withhold their location.

Homepage administration

A homepage is a dashboard that consists of navigational elements, functional controls, and system information.

When a user logs in to an instance, the default homepage defined for their role appears unless the user switched to another homepage. Administrators can customize several settings for homepages, including settings that control read or write access for any homepage and how homepages render.

Administrators can also perform the following actions for homepages.

- Create new global homepages for all users or specific homepages for users with a specified role.
- Specify an alternate login landing page that appears instead of a homepage.
- Configure the homepage splash page, which is a lightweight page that loads before a homepage loads.

Restrict content additions to a homepage

By default, anyone with a role can add content to a homepage.

If the user does not have permission to edit the homepage, the edits are applied to a homepage specific to the user. You can restrict who can add content to homepages by modifying the following property under the **Homepage Admin > Properties** module.

**Note:** The option **Add to homepage** does not take into account the property cms.glide.add_content. This means that users can add reports to any homepage they can view. The result is that users create new homepages with the added information when they modify a homepage that they don't own.
Secure a homepage

Homepages have two types of roles: read and write. Read roles limit who can view the page, and write roles limit who can make edits to the page, such as moving around gauges or deleting the homepage.

Role required: admin

**Note:** If no write roles are specified on a homepage, then any user with one of the specified read roles may access this page and delete it.

**Note:** The option *Add to homepage* does not take into account the property `cms.glide.add_content`. This means that users can add reports to any homepage they can view. The result is that users create new homepages with the added information when they modify a homepage that they don’t own.

1. Navigate to **Homepage Admin > Pages**.
2. Select the homepage you want to secure.
3. Click the edit icons next to Write roles or Read roles.
4. Move the roles you want to restrict homepage access to from the Available column to the Selected column.
5. Click **Done**.
6. Click **Update**.

Homepage user preferences

You can set your homepage user preferences.

After users have selected a homepage to view, their homepage preference is stored in **User Administration > User Preferences** in a preference named homepage. The value of the preference is the sys_id of the homepage the user sees. You can access a list of homepages by navigating to **Homepage Admin > Pages**.

Users must have at least one role to view homepages.

Add a homepage to an update set and application

You must manually add homepages and content pages to update sets prior to distribution.

Role required: admin
After inclusion, the update set always uses the current version of the page and does not require additional action. Applications automatically include any homepages and content pages that are created within or associated to an application.

1. Navigate to **Homepage Admin > Pages**.
2. Right-click a homepage record.
3. Select **Unload Portal Page**.

   The page is added to the current update set.

### Add existing gauges to a homepage

A gauge is visible on a homepage and can contain up-to-the-minute information about current status of records that exist on tracked tables.

You can add gauges to a homepage. Gauges can be created from reports or from lists of records on a table and can be sorted and filtered as needed.

1. Navigate to a homepage.
2. Click the add content icon (+)
3. Select **Gauges** in the top left column of the **Add content** pop-up window.
4. Select a table in the top middle column.
5. Select an available gauge in the top right column.
6. Click the appropriate **Add here** button to place the gauge in the corresponding section of the homepage.
7. To add more gauges, repeat steps 3 through 6.
8. Click the close icon (×) in the top right corner when you finish adding gauges.

### Arrange gauges on a homepage

Arrange gauges on custom homepages to customize your ServiceNow experience. You can always restore the homepages to the arrangement your system administrator created.

- **Moving gauges** - Click, hold and drag near the top of a gauge to move it another area. Gauges in the top or bottom areas display at 100% of the available width. Gauges in the right, left, or center areas display at 50% of the available width. If gauges are added to all these areas, they display at 33% of the available width.

- **Removing gauges** - Click the section close icon (×) to remove a gauge.
- **Refresh** - Refresh the data in a gauge by clicking the refresh icon at the upper right corner.

### Create a new gauge from a report

You can create any report as a gauge and then add that report to a homepage.

**Role required:** gauge_maker or admin

Gauges created from list reports do not preserve user list view preferences, such as the option to automatically expand grouped records.

1. Navigate to **Reports > View / Run**.
2. Open or create a report that you want to access from a gauge.
3. Open the **Save** menu and select **Make Gauge**.

   The page refreshes with a message that the gauge was created from the report.
4. Reopen the **Save** menu and select **Add to Dashboard**.

*Create a new gauge from a list of records*
You can create gauges from lists of records, for example My Work and Unassigned Incidents.

Role required: gauge_maker or admin

1. Navigate to **System UI > Gauges** and click **New**.
2. Enter a unique name for your gauge.
3. In the **Type** field, select **List**.
4. Select a table.
5. In the **Aggregate** field, select **Count**.
6. In the **Query** field, define the conditions that records must meet to be included in the gauge.
   Other fields on the form are not used when creating a list gauge.
7. Click **Submit** to save the gauge, or click **Try It** to save and view the new gauge.

After you save the gauge, you can *add it to a homepage*.

*URL gauge variable height workaround*
URL gauges can be used to display a different web page or website as part of a homepage.

URL gauges are set with a fixed height, meaning that you may have to scroll to see the entire web page in the gauge. URL gauge widths are controlled by the positioning of the gauge on a particular homepage. You cannot modify URL gauge height, but you can set up an iFrame in a UI page to manage the height.

1. First, create a UI page. The UI page name must start with `render_gadget_` (for example, `render_gadget_iFrameSNC`). Add the following to the **HTML** field, replacing the `src` with the URL that should appear in the iFrame. The height you specify must be a static value and you must use CSS style tags for this to work.

   ```html
   <iframe id="myframe" src="http://www.service-now.com" scrolling="yes" style="height:450px; width:100%"></iframe>
   ```

2. Next, navigate to **System UI > Widgets** and create or modify a widget to use for selecting your UI page from the **Add content pop-up window** on a homepage. The widget script must reference your UI page. The widget **Name** must be the name of the category in the **Add content** pop-up window. For **Renderer type**, select **JavaScript**. The following script can be added to the **Script** field if you want to create a new widget.

   ```javascript
   function sections () {
        return { 'ServiceNow' : { 'type' : 'iFrameSNC' } };  
    }

    function render ( ) {
        var scope = gs.getCurrentScopeName();
        var type = renderer.getPreference('type');
        return renderer.getRenderedPage(scope + type);  
    }

    function getEditLink ( ) {
        var scope = gs.getCurrentScope();
        var type = renderer.getPreference('type');
        return "sys_ui_page.do?sysparm_query=name=render_gadget_" + scope + type;  
    }

3. When the widget is saved, go to any homepage and add the new content.
*Count gauge colors*

In a count gauge, the count item color is based on the upper and lower limit values on the gauge panel.

The following colors are used:

- **Red**: If the count result is greater than or equal to the upper limit.
- **Orange**: If the count result is between the upper and lower limit.
- **Green**: If the count result is less than or equal to the lower limit.

You cannot customize these colors.

The upper and lower limit fields are in the Gauge [sys_gauge] table. You can configure the Gauge form to add these fields.

**Top Searches homepage**

The Top Searches homepage displays the most popular searches over the past hour, day, week, or month. The homepage includes widgets for specific tables and across the entire instance.

- Users with the itil role can view the homepage by default. To control access to the page, administrators can secure the homepage.
- Users can add any of the top searches widgets to their homepages. Administrators can control who has access to top searches widgets.
- Users can select the time period for which to view top searches and perform searches directly from the widget.
Figure 158: Top Searches homepage
Use a top search widget
Top searches widgets display the top 10 searches over the past hour, day, week, or month for the specified table or across the entire instance (All).

Take either of the following actions to use a top searches widget.

- Click the desired time period for which to see top searches (default is Day). You can change the time period at any time, and your most recent selection is loaded the next time you view the homepage.
- Click any item on the list to search for it. The Top Searches - All widget performs a global search and a table-specific widget performs a search of the table or application. For example, knowledge searches return knowledge base search results.

Note: Each user's time period selection is stored by the top_searches.period user preference.

Add a top search on a homepage
You can add a top searches widget to a homepage.

1. Click the add content icon (+) in the top left corner.
2. Select Text Search from the left panel and Top Searches from the middle panel.
3. Select the desired top search widget from the right panel.
4. On the bottom of the window, click Add here in the appropriate layout position, and then close the window.
5. Select the time period for which you want to see the top searches (default is Day).
   The widget displays the top 10 searches for the selected time period.

Control access to a top search widget
You can control who has access to top searches widgets by restricting who can add content to homepages or by applying roles to the widget.

Role required: admin

1. Navigate to System UI > Widgets.
2. Select Text Search.
3. Click the edit user roles icon and select the required access rights, then click Done.
4. Click Update.

Define a table for a top search
You can define the tables for which top searches widgets are available.

Role required: admin

1. Navigate to System Properties > Text Search.
2. Enter the tables for which you want top searches widgets in the Comma-separated list of table names which will have home page top searches widgets available for them field (the glide.ts.widget.top_search_tables property).
   The default tables are Knowledge, Problem, Incident, Change Request, Catalog Item, and Live Message (kb_knowledge,problem,incident,change_request,sc_cat_item,live_message).

   Note: Adding a parent table includes all searches on tables that extend the parent table. For example, if you add Task, then all Incident, Problem, and other task tables are included.
3. Click Save.
View user preference for time period
Each user's time period selection is stored with a user preference.

- **top_searches.period**: stores the user's selection for the Top Searches - All widget.
- **top_searches.period.<table name>**: stores the user's selection for the Top Searches - <Table> widget. For example, **top_searches.period.kb_knowledge** stores the user's preference for Top Searches - Knowledge.

1. To view user preferences, navigate to User Administration > User Preferences.
2. Find the top searches preferences.
3. Make the necessary changes to the user preferences and save the record.

Update a top search statistic
Top search statistics are compiled on an hourly basis by the TS Search Summary scheduled job.

Role required: admin

Use the following procedure to manually refresh the list.

1. Navigate to System Scheduler > Scheduled Jobs > Scheduled Jobs.
2. Open TS Search Summary.
3. Change the date and time in the Next action field to run the scheduled job at the specified time. For example, if it is currently 9:48 AM and you want to run it immediately, set the date to today and the time to 10:00.

   **Note:** This scheduled job is designed to collect hourly data. For optimal data accuracy, do not run this job more frequently.

4. Click Update.

Custom homepage widgets
There are a number of widgets provided out-of-box to be added to a homepage, but administrators can also create their own widgets.

To create a widget, an administrator must first create a UI page to control its functionality. UI pages are coded in either HTML or Jelly script, and will power the functionality of the widget.

Creating a UI page is outside the scope of this article. However, once the UI page exists, it will be used to render the widget. It is very easy to create a trivial UI page named “my_ui_page” with a single line of text, add it to a homepage, and see how it renders.

After the UI page is scripted, create a widget that renders the UI page and executes its script.

Create a widget referencing the UI page

Role required: admin

   **Note:** This functionality requires a knowledge of JavaScript.

1. Navigate to System UI > Widgets and click New.
2. Give the widget a **Renderer type** value of JavaScript and a **Name** value of My Custom Page.
3. Include the following required functions in the script.

   - The **sections()** function adds the widget to the **Add Content** section of a homepage.
The `sections()` function needs to return a JavaScript array describing how the widget should appear in the homepage Add Content UI, and what UI page should be displayed. Here is an example:

```javascript
function sections() {
    return {
        "Cool Page" : { "name" : "my_ui_page" }
    };
}
```

The above example appears in the Add Content UI as shown below, where the Name of the widget appears in the first column, and the Cool Page text appears in the second column. The UI page displayed in the preview is my_ui_page:

- The `render()` function defines how the widget should be rendered in the homepage. It can be the same for all custom widgets. This code works with scoped and global applications:

  ```javascript
  function render() {
      var scope = gs.getCurrentScopeName();
      var page = renderer.getPreference('name');
      return renderer.getRenderedPage( scope + page);
  }
  ```

  The function uses the value of the name object returned by the `sections()` function. In the example, the value is my_ui_page. This is the name of the UI page that is rendered. If the UI page does not exist, then Form not found is displayed.

- The `getEditLink()` function defines the target page to use when editing the widget. It can be the same for all custom widgets. This code works with scoped and global applications:

  ```javascript
  function getEditLink() {
      var scope = gs.getCurrentScopeName();
      var page = renderer.getPreference('name');
      return "sys_ui_page.do?sysparm_query=name=render_gadget_" + scope + page;
  }
  ```

  The function uses the value of the name object returned by the `sections()` function. In the example, the value is my_ui_page. This is the name of the UI page that is rendered. If the UI page does not exist, then Form not found is displayed.
Here is the entire example widget with all functions.

Create a new personal homepage

Users with at least one role can create new personal homepages that are visible to themselves.

You can create a completely new personal homepage with no content on it, or you can create a new homepage that uses the content of an existing page. Either way, the personal homepage is only visible to you by default unless the administrator makes your homepage global.

1. Navigate to **Self-Service > Homepage** to open your current homepage.
2. In the **Switch to page** choice list in the header, select **New page**.
Your new homepage appears with a single dropzone.

3. **Add content** as needed.

Create a personal homepage based on an existing homepage

Users with at least one role can create personal homepages based on existing homepages that are visible to themselves.

1. Navigate to any existing homepage that you have access to.
2. **Change the layout** of the homepage or add new content.

   The system automatically creates a new homepage with the word My in front of the title, indicating that the homepage is a personalized one. Any changes you make are applied to the personalized version, not the global version.

Create a global homepage or a homepage for specific users

You can create and edit a global homepage that is accessible to all users, or homepages that are accessible to users with specific roles.

Role required: admin

1. Navigate to the **Homepage Admin > Pages**.
2. Click **New**.
3. Complete the form.

### Table 116: Portal page fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Enter the title of the homepage that appears at the top.</td>
</tr>
<tr>
<td>Selectable</td>
<td>Select this option to make the homepage appear in the <strong>Switch to page</strong> choice list for users. If you specify a user or a role, only the specified user or the users with the specified role can see the page in the choice list.</td>
</tr>
<tr>
<td>View</td>
<td>Enter a unique value that you can use when referencing a global homepage in a URL or module. The View value must be unique. Otherwise, if the View is used by multiple homepages, the Edit Homepage link may not redirect to the current homepage record. You must specify a view to be able to edit the homepage for all users who can access it.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| User        | If you want the homepage to be a personal homepage for a specific user, select it here. Clear this field to keep the homepage global or if you want to specify the roles that are required to access the homepage.  
  **Note:** If this field is populated, the homepage cannot be global.                                                                                     |
| Order       | Enter a value that specifies where in the choice list of homepages that this page appears. Lower values move the page up the choice list.                                                                          |
| Write roles | Specify the roles you want users to have to edit this homepage.                                                                                                                                               |
  **Note:** If no write roles are specified on a homepage, then any user with one of the specified read roles may access this page and delete it. See Secure a homepage on page 477. |
| Read roles  | Specify the roles you want users to have to access this homepage. The **User** field takes precedence over this field. If you specify a user in the **User** field, only that user can view the homepage, regardless of what roles you add here and regardless of what role that user has.  
  **Note:** If no write roles are specified on a homepage, then any user with one of the specified read roles may access this page and delete it. See Secure a homepage on page 477. |

4. Right-click the header and select **Save**.
5. Click the **Edit Homepage** related link to see the homepage.
6. **Add content** as needed.

**Note:** You must click **Edit Homepage** to make changes to a global homepage that take effect for all users who can access the homepage. If you click **View Homepage** and make changes, a personal homepage is automatically created for you and those changes take effect only on that personal homepage. To see the **Edit Homepage** link, you must enter a value in the **View** field.
Table 117: Related links

<table>
<thead>
<tr>
<th>Related Links</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Homepage</td>
<td>Edit the homepage as a global homepage. Any changes you make when editing the homepage are visible to all users who can access it.</td>
</tr>
<tr>
<td>View Homepage</td>
<td>View the homepage as a user would. Any changes you make when viewing the homepage automatically create a personalized version of the homepage.</td>
</tr>
</tbody>
</table>

**Note:** You can also add the homepage as a module link in the application navigator.
The new homepage is available in the **Switch to Page** choice list for the users you specified or for all non-ESS users if the homepage is global.

### Homepage customization

Users with at least one role can see a customized homepage when logging in.

**Note:** If no write roles are specified on a homepage, then any user with one of the specified read roles may access this page and delete it. See *Secure a homepage* on page 477.

The content on the homepage comes from various sources, including:

- Graphs and charts generated from a report
- Application modules
- A service catalog category
- The scrolling news widget (the News knowledge category)

**System Administration homepage**

The System Administration homepage shows administrative tasks as catalog categories that provide administrators with quick access to all the items they need to configure core features, such as homepages, user administration, and email notifications.
Administrators can customize the System Administration homepage and add a homepage to an update set and application.

View your homepage
When users log in to an instance, their homepage appears.

Users who customized their homepage see the customized homepage. Users who did not customize their homepage see the homepage for their role, such as admin or itil. If their role has no homepage, a blank page appears.

After logging in, users can return to the last homepage they viewed by navigating to **Self-Service > Homepage**.
Note: Users see the homepage with the lowest Order value of the pages they have roles to see when they log in. For more information, see *Create a new personal homepage* on page 485.

Refresh the homepage
Homepage users can specify a refresh time of 5, 15, 30, or 60 minutes, or no refresh.

The default is Off (no refresh). Users can click the refresh icon ( ) at any time.

![ITIL Homepage](image)

**Figure 160: Home page refresh**

Modify the number of records displayed in a list gauge
List gauges on a homepage display the same number of records as you have configured for other lists.

For example, if you display 100 records in lists, a maximum of 100 records appears in list gauges. For instructions on changing this number, see *Customize the number of list rows per page* on page 67.

*Customize a homepage*
You can customize the default homepages that appear for each user.

Role required: admin
For pre-Geneva Patch 8 instances, customizations to out of the box homepages are lost the first time the instance is upgraded.

- Administrators can customize the default homepages that appear for each user by navigating to **Homepage Admin > Pages** and selecting the homepage.
- Any user who has a role can customize the homepage that is specified for their role by navigating to **Self-Service > Homepage**.

Homepage items
You can add different types of objects or widgets to homepages using the **Add content** button.

You add an item to a homepage by clicking the add content icon ( ) at the top, left.

In the pop-up window, you select the homepage item you want to add. Some items pertain to specific features and applications, such as *CMS content blocks* and the Work Management dispatch map. The available items vary depending on what is active on your system.
Warning: Each gauge on your homepage is the equivalent of running a report. For example, a homepage with 10 gauges runs 10 separate reports each time it refreshes. Keep this in mind when adding content to your homepage. If your homepage consistently loads slowly, try removing gauges to determine why.

You can reposition items on a homepage by clicking the item header and dragging it to the new location. To remove a homepage item, click the close icon (X) on the right side of the item header.

Table 118: Homepage Items

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gadgets</td>
<td>Items such as a knowledge search field and sticky notes.</td>
</tr>
<tr>
<td>Knowledge Base</td>
<td>Knowledge base categories.</td>
</tr>
<tr>
<td>Labels</td>
<td>Labels that you use to organize records.</td>
</tr>
<tr>
<td>System Applications</td>
<td>Applications and modules.</td>
</tr>
<tr>
<td>World Clocks</td>
<td>Any available world clock.</td>
</tr>
<tr>
<td>Gauges</td>
<td>All available dashboard gauges. Gauges are defined in System UI &gt; Gauges and can be configured to link to an existing report.</td>
</tr>
<tr>
<td>Catalog Categories</td>
<td>Active service catalog categories. The homepage uses the category's desktop icon as the homepage icon.</td>
</tr>
<tr>
<td>Catalogs</td>
<td>Service catalogs, including all of their active categories.</td>
</tr>
<tr>
<td>Diagnostics</td>
<td>Information you can use for monitoring and troubleshooting the instance, like database connections and JVM garbage collection.</td>
</tr>
<tr>
<td>Live Feed</td>
<td>Your company feed.</td>
</tr>
<tr>
<td>Text Search</td>
<td>Lists of top searches in the instance.</td>
</tr>
<tr>
<td>Filters</td>
<td>System definition filters.</td>
</tr>
</tbody>
</table>

For more information on adding content to homepages, see Add existing gauges to a homepage on page 478.

Edit widget appearance

You can edit the way a homepage item, also called a widget, looks.

1. Move the pointer to the upper right corner of the widget to show the editing controls.
2. Click the edit widget icon ( ) on the right side of the item header.
3. Select one or more options to control the appearance of the widget, such as to show a border, set title alignment and size, and set the report height.
4. Optional: Use the Interactivity options to make the widget a subscriber or publisher. Interactive publishers require Performance Analytics premium.
5. Click Done.
Change the homepage layout

You can change the layout of widgets on the homepage.

1. In the Switch to page list at the top right of the homepage, select Change Layout.
2. Select the layout you want to use.
   Administrators can create or modify layouts by navigating to Homepage Admin > Layouts.
3. Click Change.

Delete a custom homepage

Deleting a customized page restores the default homepage for that user.
Warning: If a user with the admin role clicks **Delete page**, it deletes the homepage from the instance, which affects all users. This action cannot be reverted.

1. Navigate to the custom homepage to delete.
2. Click the **Delete page** link in the bottom right.

3. Click **OK** to confirm the deletion.

### Homepage and content page layouts

Layouts are UI elements that define how blocks of content can be added to a page. Layouts are used for homepages and content pages. Layouts provide the following functionality.

- Administrators can define custom layouts for homepages or content pages.
- Users who are allowed to modify their homepage can select a layout for their homepage.
- Users who are allowed to add content to their homepages can choose where in the layout to add content.

Most successful web sites use a fairly limited number of layouts to define the structure of a much larger number of pages. For example, a web site may have several dozen pages, but only two or three layouts.

**Layout concepts**

Layouts define the overall structure of the page by arranging where dropzones appear on the page.

The dropzone is where content blocks, which make up the content of the page, can be added. The name "dropzone" comes from the fact that content blocks can be dropped into any of these zones. Most content blocks match the width of the dropzone, so the dropzone controls both the location and sizing of the content block.

Defining a custom layout requires knowledge of Jelly script, but there are a number of out-of-box layouts that can be used. After a layout is chosen, the layout dictates where the dropzones on a page appear.

Content blocks can be added at any dropzone to create the actual content of the site. Every layout divides the page into one or more logical areas. A number of commonly used predefined layouts are available, but you can also define your own custom layouts.
Custom layouts
A homepage layout is a UI macro whose name begins with layout_. You can create a UI macro to define a custom layout.

To view the list of available layouts, navigate to **Homepage Admin > Layouts**.

- Layouts are XHTML scripts that contains one or more dropzones.
- Dropzones are defined by a TD (table data cell) within the HTML where content can be placed.

**Note:** Knowledge of Jelly is required to understand the following example.

For example, here is the code for a layout.

```xml
<?xml version="1.0" encoding="utf-8"?>
<j:jelly trim="true" xmlns:j="jelly:core" xmlns:g="glide"
xmlns:j2="null" xmlns:g2="null">
<table border="0" cellspacing="6" id="${jvar_name}" width="100%">
<tr>
<td colspan="2" id="dropzone0" dropzone="true"/>
</tr>
<tr>
<td id="dropzone1" dropzone="true" valign="top" width="50%"/>
<td id="dropzone2" dropzone="true" valign="top" width="50%"/>
</tr>
<tr>
<td colspan="2" id="dropzone999" dropzone="true" valign="top"/>
</tr>
</table>
</j:jelly>
```

**Rules for scripting layouts:**

1. A layout must contain at least one table.
2. All dropzones must be TDs within a table.
3. A dropzone has two key attributes:
   a. Its ID starts with "dropzone" (e.g., "dropzone2" or "dropzone3").
   b. It has an attribute of dropzone="true."
4. Dropzone IDs must be unique (e.g., you can't have two dropzones named "dropzone1").

**Optional layout features:**

- Nested tables are allowed, so it is possible to have a table within a TD within a table.
- There can be TDs that are not dropzones.
- Styles and formatting elements are acceptable.
- Fixed-width TDs (as opposed to variable width) are allowed, although some content, such as graphs, must have a significant minimum width.
- Fixed-height TDs are allowed.

**Layout hints and tips:**

If cellspacing and cellpadding are set to zero, the drag-and-drop algorithm may have trouble distinguishing one cell from another, since multiple empty cells can potentially occupy the same point in space. It is still possible to place things via
the layout control, and they will render properly. However, to use drag-and-drop to move things around, do not drop both cellspacing and padding to zero.

Choose a layout
Certain users have rights to modify their homepage. For example, an ESS user cannot change the layout of a homepage, but an ITIL user can change it.

Role required: any role that lets the user change homepage content.

1. On the homepage, click **Switch to page** in the top right corner and select **Change Layout** at the bottom of the choice list.

   The dialog box that appears contains a list of available layouts and a short description of each. Select a layout to see a simplified preview of the layout.

2. Select a layout and click **Change**.

   **Note:**
   
   1. If the page already has content and the layout changes, the existing content stays in the same dropzones it started in. For example, content in dropzone2 of the old layout appears in
dropzone2 of the new layout as well, although the dropzone may be in a significantly different location on the page.

2. If the new layout has fewer or different dropzones than the old layout, there may be orphaned content. For example, if there is content in dropzone5 of the old layout, and the new layout does not have a dropzone5, that content is now orphaned. Any orphaned content is moved to dropzone1.

---

**Add content to a homepage**

The Add Content interface allows you to specify into which dropzone you want to place your content.

You can see this in action by selecting the add content icon ( ) in the top left of your homepage.
When you select a piece of content, you see a preview of the content in the middle of the dialog. At the bottom of the dialog, you see a model of your current layout with an Add here link in each dropzone.
To add content to a particular dropzone, click **Add here** in the appropriate dropzone. Your content is automatically added without closing the dialog. You see the homepage change in the background.

Users who can change content on a homepage can search for gauges within a category when adding content to a homepage, and configure the appearance of gauges on homepages.

### Homepage caching

Complex, frequently visited homepages can slow down instance performance by consuming system resources. Homepage caching can improve performance by using less resources to load homepages.

When this feature is enabled, individual homepage widgets are cached and can be used by multiple users until the underlying data they depend upon changes, requiring a re-render.

Administrators can configure properties for homepage caching.

#### Configure homepage cache properties

The render cache is enabled by default with a set of sensible defaults. However, the configuration can be modified by a system administrator based on the needs of a particular customer environment.

This behavior is controlled by a set of three properties available under **Homepage Admin > Properties**.

<table>
<thead>
<tr>
<th>Enable the homepage render cache</th>
<th>Enables or disables the entire cache. If you disable the cache, then nothing is cached and nothing is retrieved from the cache.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum amount of time an entry stays in the render cache (in seconds):</td>
<td>By default, the system pulls an entry out of the cache if its underlying data changes. For example, if you have a gauge of &quot;incidents by priority&quot; that goes in the cache, the system removes it from the cache if the underlying incident data is updated. However, even if the underlying data has not changed, the system automatically expires cache entries older than this age. The default is 60 seconds, meaning that, at most, a cache entry persists for 60 seconds before being expired.</td>
</tr>
</tbody>
</table>

| Enable aggressive caching | If aggressive caching is turned on, then the only time entries are removed from the render cache is |

---

© 2017 ServiceNow. All rights reserved. 499
if they expire. Even if the underlying data changes, the system still returns the previously rendered chart/graph/whatever until it expires from old age.

Setting this to true can significantly improve performance, but it does so at the expense of serving known stale data. Under some circumstances, however, a combination of aggressive caching and a short maximum cache age (see above), may be appropriate.

**Homepage caching notes and limitations**

1. Customers with less than one hundred or so concurrent users should not expect to see a significant benefit to this feature in most real-world scenarios. Homepage visits in these situations are infrequent enough that most widgets will have aged out of the cache before a second user visits a homepage. If you're not seeing performance problems with your homepages, tweaking the behavior of this cache is probably going to be counterproductive, or at least a poor use of your time.

2. The homepage cache is segmented by company, domain, and roles, so customers with different rights and visibility into the system cannot see each other's data by pulling it out of the cache. This does, however, mean that cache efficiency will be lower for customers making extensive use of domain separation.

3. In an environment with a cluster of multiple UI nodes (as opposed to a UI node + worker configuration), each node has its own homepage cache. Entries are removed from the homepage cache only when their data changes on the node upon which they are cached. Entries are, of course, still removed from the cache when they expire.

**Homepage splash page**

When the homepage splash page is configured, a lightweight splash page loads before a homepage loads. The result is a more responsive experience for users.

For example, after logging in, users can navigate directly to the information they need without waiting for the homepage to load.
Figure 162: Homepage splash page

Activate the homepage splash page
Homepage splash pages are automatically activated for new instances. If you are upgrading from a previous version, activate the Homepage Splash Page plugin if it is not already active.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Configure a property for a homepage splash page
The Homepage Splash Page plugin installs the glide.home.page property and modifies the value of the glide.login.home property.

1. Enter sys_properties.list in the navigation filter and press the Enter key.
2. Select the glide.home.page system property.
   This property defines which page to load when a user selects a homepage from the banner.
3. Set the Value field to one of the following strings.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To use the homepage splash page (default)</td>
<td>home_splash.do?sysparm_direct=true</td>
</tr>
</tbody>
</table>
Manage a service catalog homepage

The homepage for a service catalog provides the primary front end for ordering items within that catalog. Administrators and catalog administrators can design a homepage in any of the following ways.

- Customizing the catalog homepage.
- Adding, removing, and arranging categories.
- Enabling content types.
- Using catalog properties to provide additional control over behavior and appearance.
- Using renderers to define the appearance of categories.

**Note:** End users can access the catalog through the customizable user homepage, or with content management pages that use content blocks for categories.

Enable content types

To select content types, also called widgets, to use for a catalog:

1. Navigate to **Service Catalog > Catalog Policies > Properties**.
2. In the **List of content types (comma-separated) to allow on the catalog homepage** (glide.sc.home.filter) property, enter the content types to use in the service catalog.

   To see a list of content types available, navigate to **System UI > Widgets**.
   
   The default value is **Service Catalog**, which includes only service catalog categories. For example, to include service catalog categories and gauges, set the property to **Service Catalog, Gauges**.

Customize a catalog homepage

1. Navigate to **Service Catalog > Catalog**.
2. Click the add content icon (+) at the top of the page to add a category.
3. Select a category in the middle panel of the pop-up window that appears.
4. Click **Add here** in the location where the category should appear on the homepage.
5. Optional: Repeat steps 3-4 to add more categories.
6. Close the pop-up window.
7. Optional: To change a category's location, drag it to the appropriate place.
8. Optional: To remove a category, click the (X) icon on the top right of the category header.

Define the mobile layout
You can configure the mobile layout for categories within a service catalog.

Role required: admin

By default, service catalog categories appear on mobile devices in the same order as on desktop devices.

1. Navigate to Service Catalog > Mobile Admin > Mobile Layout to display a list of service catalogs.
2. Click the lookup icon for the service catalog you want to configure.
   The mobile layout details for that catalog displays.
3. Click Edit to select which categories appear on mobile devices, and in which order.
4. Add, remove, or reorder the selected mobile categories, and then click **Save**.

**Note:** To return the display to the default desktop layout settings for the portal page associated with that catalog, click the **Replace categories with desktop layout** related link.
Configure content types
You can configure content types for the service catalog.

You can use service catalog properties to configure content types, which are also called widgets.

Navigate to Service Catalog > Catalog Policies > Properties to view service catalog properties.

Navigate to System UI > Widgets to see a list of content types available.

Enable content types for the Service Catalog

By default, only service catalog categories are displayed on the service catalog homepage.

To select additional content types, add these types in a comma-separated list in the List of content types (comma-separated) to allow on the catalog homepage property (glide.sc.home.filter).

For example, to include service catalog categories and gauges, set the property to Catalog Categories, Gauges.

Display the expand / collapse icon

By default, the expand / collapse icon is not displayed for categories on the service catalog homepage.

To display this icon on the left of each category, check Yes beside the Toggle whether the expand/collapse icon is rendered for category widgets on the service catalog homepage property (glide.sc.homepage.show.collapse).
Configure search and navigation
You can use service catalog properties to configure search and navigation within the service catalog homepage.

Navigate to Service Catalog > Catalog Policies > Properties to view service catalog properties.

Restrict search access
Each catalog homepage provides a search bar to help locate items not displayed on the homepage. By default, any user who can access the catalog homepage can search using this search bar.

To restrict access to this search function by role, use the List of roles (comma-separated) that can search the service catalog property (glide.sc.can_search).

For example, to only allow logged-in users to access search, set this to blank. To remove the search function for all users, set this to None.
Note: Search results only show items that the logged-in user can access.

**Hide search results from inactive categories**

By default, search results are returned from all categories, including inactive categories.

To avoid returning results from inactive categories, set the **Service catalog searches return items in inactive categories** property (glide.sc.search.disabled_cats) to **No**. For example, when designing a new category, you may want to avoid users seeing search results from this category until you are ready to make it active.

Note: Security constraints may also make a category or catalog item inaccessible.

**Restrict search by item type**

By default, a search field does not appear when viewing order guides, record producers, and wizard launchers.

To define which catalog item types do not show the search field, list these types in the **List of class names for catalog items that do not have the search field displayed** (glide.sc.item.cannot_show_search) property.

For example, to enable searching when viewing record producers, remove `sc_cat_item_producer` from this list.
Request a reset of a password for a service or an application.

Request a reset of a password for a service or an application.

Whose password needs to be reset?

Joe Employee

What application password do you need reset?

How would you like to be contacted with your new password?

More Information

- Email
- Telephone
- SMS

Submit

Figure 164: Service catalog search bar removed
Enable breadcrumb links

By default, breadcrumbs appear without links when using content management pages as service catalog homepages.

To display these breadcrumbs with links, set the **Use links for breadcrumbs rendered in Service Catalog pages accessed via a CMS site** property (glide.sc.use_breadcrumb_links.cms) to Yes. This provides greater navigational control for end users.

Disable search suggestions

By default, service catalog searches display “Did you mean?” suggestions if a search does not return any results and an alternate spelling or similar recent search does.

To disable these suggestions, set the **Specify whether search suggestions should be enabled** property (glide.sc.search.suggestions) to No.

Change the parameter for parallel homepage rendering

The system uses a parallel rendering algorithm to speed up home page rendering.

Role required: admin
In testing, the average performance benefit is between 20% to 50% depending on the page in question. Administrators can set parallel homepage rendering properties for optimal performance on an instance. To change the parameters for parallel homepage rendering, add the following system properties.

1. glide.ui.homepage.parallel: enables or disables parallel homepage rendering. The default value is true.
2. glide.ui.homepage.parallelism: sets the number of threads to use when rendering a homepage. The default value is 2.

Turn on homepage render time

To see render times on homepage widgets, you can turn on homepage debugging.

Role required: admin

Navigate to System Diagnostics > Debug Homepage Render.

For the remainder of your session, load times appear on homepage widgets and debugging messages appear at the bottom of homepages. Use these times to identify items that may slow down homepage rendering.
Debug Output  Others

13:24:53.431: Homepage Widget "Users by Location" output in time: 0.108
13:24:53.441: Homepage Widget "My Groups Work" output in time: 0.120
13:24:53.583: Homepage Widget "Open Items by Escalation" output in time: 0.130
13:24:53.596: Homepage Widget "My Work" output in time: 0.157
13:24:53.610: Homepage Widget "ITIL Summary Counts" output in time: 0.020
Troubleshoot a report on a homepage

You can troubleshoot reports that may be impacting homepage performance by identifying which reports are on the problem homepage, determining which reports are running slowly, and correcting problems in slow reports.

Role required: admin

1. Navigate to **Homepage Admin > Pages**.
   
   Opening the homepage record instead of the homepage saves time and system resources.

2. Open the problem homepage record.

3. In the **Portal** related list, note item names in the **Summary** field.
   
   These items may not all be reports.

4. Enter `sys_report.list` in the navigation filter to open a list of all reports.

5. Search for a report title that matches an item on the homepage that you noted down from the **Summary** field.

6. Open the report and run it.

7. If the report runs slowly, look for and correct the following common reporting mistakes.

   • Returning too many results
   • Grouping by fields such as duration or name
   • Reporting on a user-created table that uses many joins on other tables

8. If the report runs slowly and correcting common mistakes does not help, examine the type of report and what data it reports. Look for ways to optimize or replace the report.

9. Repeat these steps for each item on the homepage.

Create a widget that displays a ServiceNow UI page

You can create a widget that displays a ServiceNow UI page. This widget can be added to homepages and dashboards.

Role required: admin

A UI page is a ServiceNow page that is not a list or a form. Certain UI pages, such as external site widgets or gadgets, may not display properly on a homepage or dashboard.

1. Identify the UI page that you want to create into a widget and note its name.

2. Navigate to **System UI > Widgets** and click **New**.

3. Enter a **Name** for the widget. This field defines the first category a user selects when adding the widget to a dashboard or homepage.

4. Enter the following code, making substitutions as noted below.

   • **Widget** is the title of the widget. This title appears on the top of the widget when you add it to a dashboard or homepage.
   • **widgetname** is the name of the UI page from step 1. This field defines the second or subcategory a user selects when adding the widget to a dashboard or homepage.

```javascript
function sections() {
    return {
        'Widget': {
            'type': 'widgetname'
        }
    };
}
```
function render() {
    var scope = gs.getCurrentScopeName();
    scope = (scope == "rhino.global" ? "" : scope + "_");
    var page = renderer.getPreference('type');
    return renderer.getRenderedPage(scope + page);
}

function getEditLink() {
    var scope = gs.getCurrentScopeName();
    scope = (scope == "rhino.global" ? "" : scope + "_");
    var page = renderer.getPreference('type');
    return "sys_ui_page.do?sysparm_query=name=" + scope + page;
}

To learn how to make a UI page without using a framework page template, see the Making a UI page without using the framework page template blog posting by a developer in the ServiceNow Community.

State flows

State flows enable an administrator to customize transitions from one state to another in tables derived from the Task [task] table and configure the system to perform work during transitions to specific states.

An example of a state transition is when the State field in an incident changes from Active to Awaiting User Info. An administrator might want to trigger an event during this transition or make a specific field mandatory when the incident reaches the end state.

State transitions in the Work Management application were reimplemented to use state flows.

Installed with state flows

Several types of components are installed with state flows.

Tables installed with state flows

Tables

State flows adds the following tables.

Table 119: Tables for state flows

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State Flow [sf_state_flow]</td>
<td>Contains state flow definitions. This table contains all state flow definitions, including those for work orders and work order tasks.</td>
</tr>
<tr>
<td>Work Order Flow [sf_work_order]</td>
<td>Contains state flow definitions for work orders. This table is installed when Work Management is activated.</td>
</tr>
<tr>
<td>Work Task Flow [sf_work_task]</td>
<td>Contains state flow definitions for work order tasks. This table is installed when Work Management is activated.</td>
</tr>
</tbody>
</table>

Business rules installed with state flows
## Business rules

State flows adds the following business rules.

### Table 120: Business rules for state flows

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assert Field Uniqueness in State Flow</td>
<td>State Flow</td>
<td>Ensures that business rules and UI actions are not accidentally copied to new state flows.</td>
</tr>
<tr>
<td>Check Client Script</td>
<td>State Flow</td>
<td>Adds a client script to new records.</td>
</tr>
<tr>
<td>Check Event Rule</td>
<td>State Flow</td>
<td>Adds or deletes event rules, as the event field is updated.</td>
</tr>
<tr>
<td>Check Work Notes Rule</td>
<td>State Flow</td>
<td>Adds or deletes work note rules, as the work notes for a state flow are updated.</td>
</tr>
<tr>
<td>Create Business Rule</td>
<td>State Flow</td>
<td>Automatically creates a business rule when automatic conditions or script are present.</td>
</tr>
<tr>
<td>Create script for Field controls</td>
<td>State Flow</td>
<td>Create scripts for field controls, when they are in use.</td>
</tr>
<tr>
<td>Create UI Action</td>
<td>State Flow</td>
<td>Automatically creates a UI action when manual conditions or script are filled in.</td>
</tr>
<tr>
<td>Delete Related Elements</td>
<td>State Flow</td>
<td>When state flows are deleted, delete all related client scripts, business rules, UI actions and overrides.</td>
</tr>
<tr>
<td>Remove script for Field controls</td>
<td>State Flow</td>
<td>If all field controls are disabled, see if any of the client scripts should be removed.</td>
</tr>
<tr>
<td>State Change</td>
<td>State Flow</td>
<td>Get the correct state choice value when the state is changed.</td>
</tr>
<tr>
<td>Update dependent records</td>
<td>State Flow</td>
<td>When a state flow is made active or inactive, ensure the business rule and UI actions are made active or inactive as well.</td>
</tr>
</tbody>
</table>

*Script includes installed with state flows*
Script includes

State flows adds the following script includes.

### Table 121: Script includes for state flows

<table>
<thead>
<tr>
<th>Script include</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateFlow</td>
<td>Implements state flows and supports creation of state flow elements, such as business rules, UI actions, dictionary overrides, and client scripts.</td>
</tr>
<tr>
<td>StateFlowAJAX</td>
<td>Allows access to state flow functionality from client scripts.</td>
</tr>
</tbody>
</table>

Use state flows

You can create custom state flows for any table that uses states.

Make state transitions occur in any order or skip specific states according to your business practices. State flows can manage the behavior of fields and field visibility on a task form. State flows can also trigger system events that perform work such as sending *email notifications*.

Administrators can configure state flows to:

- Limit the *choice list* for the *State* field to contain only those states that are valid for the flow.
- Define the *behavior and visibility* of specific fields on a task form when state transitions occur and for certain end states.
- Configure *events that are triggered when a task changes states*.
- *Set up* UI actions and business rules to implement certain state transitions, or while the task is in certain states.

The State Flow plugin installs records used by work management to control state transitions for work orders and work order tasks.

Creating and customizing state flows requires scripting knowledge. Users with the admin role can create state flow records.

Create a state flow

Creating State Flows.

1. Navigate to *State Flows* > *State Flows* and click *New*.
2. Fill in the fields, as appropriate.

   The system enforces the field controls with the same client script that filters the choice list for the State field.

### Table 122: Creating state flows

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Automatically generated record number</td>
</tr>
<tr>
<td>Table</td>
<td>[Required] Table on which the state flow record runs. Only tables that extend the Task [task] table are available in the list.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Starting state</td>
<td>Name of the state at the beginning of the transition. The selections in this field are filtered by the possible states for the table selected.</td>
</tr>
<tr>
<td>Ending state</td>
<td>Name of the state at the end of the transition. The selections in this field are filtered by the possible states for the table selected.</td>
</tr>
<tr>
<td>Client script</td>
<td>Client script to run for this transition. The client script controls the available states you can select by limiting the contents of the State choice list to valid states. This client script also controls specific field behavior configured for state changes in the Field Controls section of the form.</td>
</tr>
<tr>
<td>Event</td>
<td>Name of an existing event to trigger when this transition occurs. See Triggering Events on State Changes for more information.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of this record. Make sure the name is descriptive of the state transition or the processing that the record is performing. This name does not have to be unique.</td>
</tr>
<tr>
<td>Roles</td>
<td>Not used for any processing.</td>
</tr>
<tr>
<td>Active</td>
<td>Enables or disables this state flow record.</td>
</tr>
<tr>
<td>Class</td>
<td>Defines the state flow class for this record. The system selects the appropriate class from these options:</td>
</tr>
<tr>
<td></td>
<td>• State Flow: Records created for state flows in all task-based tables except those in work management.</td>
</tr>
<tr>
<td></td>
<td>• Work Order Flow: Records created for state flows in the Work Order [wm_order] table. This class is available when work management is activated.</td>
</tr>
<tr>
<td></td>
<td>• Work Task Flow: Records created for state flows in the Work Order Task [wm_task] table. This class is available when work management is activated.</td>
</tr>
<tr>
<td>Dictionary override</td>
<td>Sets the starting value for the State field on all new records for the table named in the state flow record. See Dictionary Overrides for configuration procedures.</td>
</tr>
<tr>
<td>Work notes</td>
<td>Noteworthy comments about this state flow transition. For details about how these notes are used, see Work Notes.</td>
</tr>
<tr>
<td>Comment</td>
<td>Details about the customized record.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Manual (Runs scripts from a UI action that require the user to click a button or related link.)</td>
<td></td>
</tr>
<tr>
<td>Manual condition string</td>
<td>Conditions for enabling a UI action that cannot be defined with the condition builder. For example, you can use this string to define UI actions for mobile devices. This condition has an [and] relationship with the condition in the Manual condition field.</td>
</tr>
<tr>
<td>Manual condition</td>
<td>Conditions for enabling a UI action that can be defined for fields in the target table. This condition has an [and] relationship with the condition in the Manual condition string field.</td>
</tr>
<tr>
<td>Manual script</td>
<td>Script that defines what the UI action does when the conditions are true. This script runs when the user clicks a button or a related link.</td>
</tr>
<tr>
<td>UI action</td>
<td>[Read Only] Name of the button that the system creates to enable this transition. The system creates the label using the same name as the state flow record that created it.</td>
</tr>
<tr>
<td>Automatic (Runs a business rule automatically when a task record is changed and updated.)</td>
<td></td>
</tr>
<tr>
<td>Automatic condition string</td>
<td>Conditions for running the business rule that cannot be defined with the condition builder, such as evaluating if the proposed transition is a valid flow. This condition has an [and] relationship with the condition in the Automatic condition field.</td>
</tr>
<tr>
<td>Automatic condition</td>
<td>Conditions for running the business rule that can be defined for fields in the target table. This condition has an [and] relationship with the condition in the Automatic condition string field.</td>
</tr>
<tr>
<td>Automatic script</td>
<td>Script that performs additional work when the condition is true. This script can do tasks such as update the date and time the transition occurred or notify someone using email when a specific state change occurs. Automatic state transitions occur when changes are made to the task record.</td>
</tr>
<tr>
<td>Business rule</td>
<td>Name of the business rule created for this transition. Two conditions must be satisfied before this business rule can run. The task must be on a specific starting state, and the Automatic condition must be true. If both of these conditions are satisfied, the business rule performs the transition requested, using the starting and ending states from the State Flow form.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Field Controls (Determines <em>field properties</em> when a record transitions between states or reaches a specific end state.)</td>
<td>Makes the selected fields required when this transition occurs, or when the end state is the current state.</td>
</tr>
<tr>
<td>Mandatory fields</td>
<td>Makes the selected fields required when this transition occurs, or when the end state is the current state.</td>
</tr>
<tr>
<td>Read only fields</td>
<td>Prevents the selected fields from being edited when this transition occurs, or when the end state is the current state.</td>
</tr>
<tr>
<td>Visible fields</td>
<td>Displays the selected fields when this transition occurs, or when the end state is the current state.</td>
</tr>
<tr>
<td>Not mandatory</td>
<td>Makes the selected fields optional when this transition occurs, or when the end state is the current state.</td>
</tr>
<tr>
<td>Not read only</td>
<td>Makes the selected fields editable when this transition occurs, or when the end state is the current state.</td>
</tr>
<tr>
<td>Not visible</td>
<td>Hides the selected fields when this transition occurs, or when the end state is the current state.</td>
</tr>
</tbody>
</table>

3. Determine if you want the state transition method to be manual or automatic and open the appropriate section on the form.
   - Manual: Click **Create UI Action** to create a button on the task form that enables users to execute the transition manually. The system uses the value in the Name field as the label for the UI action. The UI action executes the script in the Manual Script field when the conditions are true. For example, a manual transition can create an **Activate** button when an incident is in the New state that enables a user to mark the incident as active.
   - Automatic: Click **Create Business Rule** to create the business rule. The business rule executes the script in the Automatic Script field when the conditions are true. For example, a business rule created by the system can set an incident state to Assigned when the Assigned to field is populated. Business rules are automatically deleted when the state flow record is deleted. For more information, see [Business Rules](#).

4. Optional: Click **Create Client Script** to create the script that limits the values offered in a task record's State field to valid states for that transition.

5. Configure the fields in the Field Controls section to control how specific fields display when a task record changes states.

*Dictionary overrides in state flows*
A dictionary override in a state flow defines the starting state for all new records in a specific table.

You set an override in tables that extend a base table only, so that your customizations are applied only to the extended table.

1. In a state flow record, select an **Ending state**.
   - This is the override value which becomes the starting state for all new records in the table named.

2. Click **Create Default Value**.
The system populates the Dictionary override field with a value of state, which is the field in the task table affected by the override. The Dictionary override field is read-only. After the override is created, the system hides the Create Default Value button on all subsequent state flow forms for that table.

**Events triggered on state changes**
You can configure a state flow to trigger a registered system event when a task transitions from a starting state to a specified end state.

For example, you can use events to trigger email notifications and create script actions. When you attach an event to a state flow, the system creates a *business rule* called State Flow Events for <table name> for the table specified in the state flow. If you specify a start and end state, the business rule executes when the record transitions from the start state to the end state. If the state flow only specifies an end state, the business rule executes whenever that end state is reached. The system creates one business rule for all state flows containing events on a single table. When all events or all state flows on a table are deleted, the system deletes the business rule.

**Field controls**
You can define controls for individual fields that are enforced when a record transitions between states.

Settings in the Field Controls section of the State Flow form enable you to apply field controls when the system detects a specified state transition or when the end state is the current state when the form is opened. The control is applied only to existing fields on the form. State flows cannot add fields to the form.

For example, you might want the Problem field to be visible when an incident moves to the Awaiting Problem state. If the incident state changes to Awaiting User Info, you hide the Problem field and make the Caller field mandatory.

The best practice when creating field controls is to configure state flow records with an ending state only and to create the correct behavior for every ending state you want to control. This ensures that the field controls are set properly when the user selects a new state, and also when the user returns a record's State field to the original state. Only specify a full state transition, with both a starting and ending state, when you want a particular behavior for that precise state transition.

**Note:** State flows use client scripts to enforce field controls. It is possible that your settings can be changed by existing UI policies, which execute after client scripts.

The system creates the following objects as needed to enforce field properties in state flows:

**Table 123: Field Controls**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business rule</td>
<td>State Flow Notes for &lt;table name&gt;</td>
<td>Enforces mandatory fields for the table on which that field behavior is defined.</td>
</tr>
<tr>
<td>Client script (onLoad)</td>
<td>&lt;table name&gt; state flow</td>
<td>Sets possible states and initial mandatory, read-only, and visible properties when a record is loaded.</td>
</tr>
<tr>
<td>Client script (onChange)</td>
<td>&lt;table name&gt; change state flow</td>
<td>Sets updated mandatory, read-only, and visible properties when a record is changed.</td>
</tr>
</tbody>
</table>

**Rebuild state flows**
You can rebuild state flows when a mismatch between existing and new sys_ids occurs.
When you use an XML file to import a state flow record into an instance, the system attempts to match the incoming states with existing states by comparing sys_ids. Because the sys_ids of items in a choice list can vary between instances, the system can fail to match the states, even though they are otherwise identical.

When matching fails, the start and end states of affected records are left blank or contain numeric values. To repair these records navigate to State Flows > Admin > Rebuild State Flows. This module runs a script that compares the numerical value of each item in the State field choice list until it finds a match in the imported state flow record.

State flow cleanup

The business rules, client scripts, and UI actions that the system creates automatically to perform custom transitions exist only while the state flow records that use them are present.

When all the state flows on a table are deleted, the system attempts to delete any unnecessary programming elements that were created on that table, using these criteria:

<table>
<thead>
<tr>
<th>Element</th>
<th>Deleted When</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI action</td>
<td>The state flow that created it is deleted.</td>
</tr>
<tr>
<td>Business rule</td>
<td></td>
</tr>
<tr>
<td>Dictionary override</td>
<td></td>
</tr>
<tr>
<td>Business rule that processes events triggered by a state flow</td>
<td>All state flows for the table specified that have events configured are deleted.</td>
</tr>
<tr>
<td>Client script (onLoad)</td>
<td>All state flows for the table are deleted.</td>
</tr>
<tr>
<td>Client script (onChange)</td>
<td>All state flows with field controls are deleted.</td>
</tr>
<tr>
<td>Work notes business rule</td>
<td>All state flows with field controls or work notes are deleted</td>
</tr>
</tbody>
</table>

Work notes

Work notes are an important part of the state flow process and are used to communicate information about state transitions.

The state flow adds these work notes to the Work notes field of any task making this transition.

These rules apply to state flow work notes:

- For a state flow with no Starting state, the work note is added every time the task transitions to the Ending state.
- For a state flow with a Starting state and an Ending state, the work note is added only when the task transitions from that starting state to that ending state.
- If two state flows with work notes have the same Ending state, but only one has a Starting state, the system adds the work notes from the state flow with the starting state. This better matches the state flow work note to the more important transition between specific starting and ending states.

Objects to enforce state flow field properties

ServiceNow creates the following objects as needed to enforce field properties in state flows.
Table 125: Objects for Enforcing Field Properties in State Flows

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business rule</td>
<td>State Flow Notes for &lt;table name&gt;</td>
<td>Enforces mandatory fields for the table on which that field behavior is defined.</td>
</tr>
<tr>
<td>Client script (onLoad)</td>
<td>&lt;table name&gt; state flow</td>
<td>Sets possible states and initial mandatory, read-only, and visible properties when a record is loaded.</td>
</tr>
<tr>
<td>Client script (onChange)</td>
<td>&lt;table name&gt; change state flow</td>
<td>Sets updated mandatory, read-only, and visible properties when a record is changed.</td>
</tr>
</tbody>
</table>

Survey Management

The Survey Management application allows you to create, send, and collect responses for basic surveys.

Table 126: Survey applications

<table>
<thead>
<tr>
<th>Survey Management</th>
<th>Survey Management improves the user interface and extends the capabilities of the Legacy Surveys application.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Note: Survey administrators can continue to use legacy survey functionality and data. The recommended course of action, however, is to migrate legacy surveys to the Survey Management platform. Concurrent use of both Survey Management and Legacy Surveys applications can result in confusion and redundancy.</td>
</tr>
<tr>
<td></td>
<td>Survey wizards are not impacted and cannot be migrated.</td>
</tr>
<tr>
<td>Legacy Surveys</td>
<td>The Legacy Surveys application is not described in the documentation that you are currently viewing. The Legacy Surveys application is documented on the ServiceNow wiki.</td>
</tr>
</tbody>
</table>

Table 127: Version comparison

<table>
<thead>
<tr>
<th>Capability</th>
<th>Surveys</th>
<th>Legacy Surveys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save new survey responses each time a user takes the same survey.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Capability</td>
<td>Surveys</td>
<td>Legacy Surveys</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>---------</td>
<td>----------------</td>
</tr>
<tr>
<td>Create question templates to reuse sets of answer options.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Categorize survey questions and report on category results.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Deactivate a survey without deleting it.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Create conditional questions.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Send surveys automatically on a schedule.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Customize survey questionnaire color scheme.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Save anonymous survey responses for logged-in users.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>View survey responses on graphical scorecards.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Save surveys in a draft state until they are ready to publish.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Create and send surveys from one page.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Allow only specific users to access a survey.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Send surveys based on conditions.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Send survey email notifications.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Limit how often a user can take the same survey.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Add introduction and end note text.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Create survey modules.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Public survey:</strong> Allow persons to take a survey without logging in.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
There are also many options for advanced configuration. Some of the many things you can do with surveys include:

- Create a survey, add questions, and choose recipients, all in one interface.
- Create conditional questions, which appear only when users answer other questions a certain way.
- Restrict a survey so only specific survey users can take it and send all of them invitations simultaneously. Alternatively, make the survey a public survey so that any user can take the survey, even users who have not logged in to the ServiceNow system.
- Set a schedule to automatically assign a survey to users and to limit how often the same user can take a survey.
- Customize the look and feel of survey questionnaires.
- Save anonymous survey responses.
- Convert survey responses to numerical scores and view them on scorecards.
- Deactivate a survey for maintenance or to retire it without deleting it.

**Note:** Because surveys use the same tables and other back-end components as assessments, you may see assessment elements such as table and field names in certain places throughout the survey feature.

### Survey Management roles

The Survey Management application uses the following roles.

No role is required to take survey questionnaires that are assigned to you.

#### Table 128: Survey management roles

<table>
<thead>
<tr>
<th>Role title</th>
<th>Role name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey administrator</td>
<td>survey_admin</td>
<td>Survey administrators create and administer surveys. They know what types of surveys are necessary, when to send a survey, and to whom. Survey administrators can use all modules in the Survey Management application menu. The survey_admin role contains the survey_reader and assessment_admin roles.</td>
</tr>
<tr>
<td>Survey reader</td>
<td>survey_reader</td>
<td>Survey readers view surveys and related information, such as survey responses, survey groups, scorecards, and reports.</td>
</tr>
<tr>
<td>Administrator</td>
<td>admin</td>
<td>Administrators have access to all aspects of the survey processes. Only administrators can modify survey notifications, create survey modules, and import surveys.</td>
</tr>
</tbody>
</table>
Key survey terms
Survey admins use particular terms when working with surveys.

Table 129: Key survey terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey definition</td>
<td>Root record on which a survey is built. A survey definition contains information like the survey name, state, and distribution schedule. Each survey definition has its own categories and questions. Survey definitions are comparable to legacy survey masters.</td>
</tr>
<tr>
<td>Public survey</td>
<td>Any user can take a public survey, even users who have not logged in to the ServiceNow system. For a completed survey, the Assigned to field value is Guest.</td>
</tr>
<tr>
<td>Survey category</td>
<td>Represents a theme for survey questions. Categories contain one or more questions. The system creates one category per survey by default. Additional categories are optional.</td>
</tr>
<tr>
<td>Survey question</td>
<td>A question that appears on survey questionnaires for the associated survey definition. Survey questions are comparable to legacy survey questions.</td>
</tr>
<tr>
<td>Survey user</td>
<td>User who is authorized to receive invitations for a restricted survey.</td>
</tr>
<tr>
<td>Survey instance</td>
<td>Represents one survey questionnaire assigned to one user. Survey instances are comparable to legacy survey instances.</td>
</tr>
<tr>
<td>Trigger condition</td>
<td>Defines a rule that enables the system to send a survey when an action occurs on a table, such as when an incident closes. Trigger conditions are comparable to legacy survey conditions.</td>
</tr>
<tr>
<td>Scorecard</td>
<td>Visual representation of survey responses. Scorecards display a variety of data summaries for one survey definition.</td>
</tr>
</tbody>
</table>

Survey results
Survey results for each question and category are calculated automatically.

If you use survey result calculations for results and scorecards, ensure that the Positive indicator field for the question is set appropriately, based on the answer options. To have any results, a category must contain scored questions.
Survey responses

Survey results are stored in the Metric Result [asmt_metric_result] table and display the recipients’ answers to each question in a category. To view general results, navigate to Survey > Survey Responses. To view results for a particular criterion, use a filter on the Metric Result [asmt_metric_result] table. For example, to view results based on the assignment group, apply a filter condition for assignment group.

Figure 166: Survey designer metric results

Category results

Category results are stored in the Assessment Category Result [asmt_category_result] table and display the overall ratings for each category based on the weighted value for each scored question. To view these results, navigate to Assessments > Results > Category results and filter the results using the [Type.Evaluation method] [is] [Survey] condition.

Figure 167: Assessment category results

Survey scorecards

A scorecard provides a visual breakdown of survey responses, based on the way questions were answered, by category. To access a scorecard, see View a scorecard on page 529.
Survey responses
Survey responses are stored on the Metric Result [asmt_metric_result] table.

There is a metric result record for each user's response to each question on every survey instance.

Survey readers and survey administrators can view results for all surveys or for one specific survey.

View results for all surveys
Survey responses are stored on the Metric Result [asmt_metric_result] table.

Role required: survey_admin or survey_reader

1. Navigate to Survey Management > Survey Responses.
2. Do not confuse this module with Survey Management > Legacy Surveys > Survey Responses, which displays legacy survey responses.
3. The Type column displays the survey definition each response is associated with. Recall that survey definitions are stored on the Assessment Metric Type table, which is what the name Type refers to.

View the results for a particular survey
Survey responses are stored on the Metric Result [asmt_metric_result] table.

Role required: survey_admin or survey_reader

1. Navigate to Survey Management > View Surveys
2. Open a survey definition.
3. Under Related Links, click View Responses, which is available only if there are results.
   - The results are grouped by metric, which is what questions are called in assessments.
4. Open a metric result to view more detail.
   - The metric result contains the user's response as well as calculated values, which advanced survey administrators may be interested in. Because the Metric Result table is also used by the assessment feature, many field names are not clear in the context of surveys.
Metric result fields
List of field descriptions for the Metric Result form.

**Table 130: Metric Result form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment group</td>
<td>Assessment group to which the metric result belongs. An assessment group is a container for survey instances and results generated in a single occurrence. The system generates an assessment group every time at least one survey instance is created. If multiple survey instances are created at once, such as when a survey administrator sends invitations to a list of survey users, they are all stored in the same assessment group.</td>
</tr>
<tr>
<td>Metric</td>
<td>Question that the user answered.</td>
</tr>
<tr>
<td>Data type</td>
<td>Data type of the question the user answered.</td>
</tr>
<tr>
<td>Method</td>
<td>Assessment method. Always Assessment for surveys.</td>
</tr>
<tr>
<td>Updated</td>
<td>Date and time the metric result was last updated.</td>
</tr>
<tr>
<td>Source</td>
<td>Survey definition from which the associated survey instance was generated.</td>
</tr>
<tr>
<td>Assigned to</td>
<td>User who completed the survey questionnaire.</td>
</tr>
<tr>
<td>Instance</td>
<td>Survey instance completed by the Assigned to user.</td>
</tr>
<tr>
<td>Actual value</td>
<td>Value obtained from the user response to the question. The actual value is determined by the question data type:</td>
</tr>
<tr>
<td></td>
<td>• Checkbox: The actual value is 0 if the checkbox is cleared and 1 if it is selected.</td>
</tr>
<tr>
<td></td>
<td>• Choice or Likert Scale: The actual value is equal to the Value of the metric definition associated with the chosen answer option.</td>
</tr>
<tr>
<td></td>
<td>• Date, Date/Time, or String: The actual value is -1 to indicate that these data types do not contribute to category result calculations.</td>
</tr>
<tr>
<td></td>
<td>• Template: The actual value is equal to the Value of the template definition associated with the chosen answer option.</td>
</tr>
<tr>
<td></td>
<td>• Yes/No: The actual value is 0 if the response is No and 1 if it is Yes.</td>
</tr>
<tr>
<td>Normalized value</td>
<td>Adjusted value that accounts for the question’s Scale definition setting, minimum and maximum values, and other factors.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>String value</td>
<td>Value that displays the response as it appears on a questionnaire. In some cases this is the same as the Actual value, such as when the question data type is Percentage. The string value is N/A for unanswered questions of certain data types.</td>
</tr>
</tbody>
</table>

**View a scorecard**

Role required: survey_admin or survey_reader

1. Navigate to Survey Management > View Surveys.
2. Open a survey definition.
   The scorecard link is hidden if there are no survey results to report.

The scorecard contains a header that displays the name of the survey and a section that displays results or compares the ratings. The scorecard presents results in the following views:

- Category Results
- Question Results
- Average Ratings
- History

**Survey scorecards**

The Survey Management application prepares printable scorecards.

A scorecard displays charts for survey results, in which category and question responses are analyzed and current ratings are compared with previous ratings. Users can examine ratings over time, compare question ratings, or compare the ratings of all categories. All ratings are averages for the time range selected. The system dynamically updates a scorecard each time you view it, so the ratings reflect recently completed surveys.

The scorecard contains a header that displays the name of the survey and a section that displays results or compares the ratings. The scorecard presents results in the following views:

- Category Results
- Question Results
- Average Ratings
- History

**Survey scorecard category results**

The Category Results view is a stacked bar chart showing survey results for all questions in a category by respondent count.

Select a category from the list to display the questions from that category in the chart.

This view displays responses that use the following data types:

- Checkbox
- Choice
- Likert Scale
- Number
- Template
- Yes/No
**Note:** The **Checkbox** and **Yes/No** data types are combined into the **Boolean** data type in the Survey Designer.
Service Desk Satisfaction Survey

Survey Scorecard

Service Desk Satisfaction Survey

Metric Result Count

© 2017 ServiceNow. All rights reserved.
To view details about a specific response, point to the colored bar to display the response, the number of responses, and the percentage it represents of all the responses to that question.

![Benefit Satisfaction](image)

**Figure 169: Survey scorecard category results detail**

Survey scorecard question results
The Question Results view shows the results for each question in a category using different chart types, based on a data type.

**Pie chart**

The pie chart shows question results for these data types:
- Checkbox
- Choice
- Likert Scale
- Number
- Template
- Yes/No

**Note:** The **Checkbox** and **Yes/No** data types are combined into the **Boolean** data type in the Survey Designer.
Service Desk Satisfaction Survey

Survey Scorecard

How courteous and respectful was the technician who responded? (1=poor, 6=excellent)

- 2 = 2 (14.29%)
- 3 = 2 (14.29%)
- 4 = 9 (54.29%)
- 5 = 1 (7.14%)
Bar chart

A bar chart appears when question results use the Percentage data type.

By default, all results for percentage questions use a report range of 20% segments. To configure a report range, navigate to Reports > Administration > Report Ranges.

Survey scorecard average ratings

The Average Ratings view displays the weighted average rating for each survey question in a category.

Use this view to learn how individual questions affect the overall rating for the category. Select a survey category from the choice list to display the chart for that category.
Service Desk Satisfaction Survey

Survey Scorecard

Average question rating in category: Service Desk Satisfaction Survey

How courteous and...  
How satisfied are...  
How satisfied were...  
Please rate the t...  
Was technician ab...  

Average Normalized value
To view the effect of each question's ratings on the entire category's ratings, point to the colored bar. The pop-up box shows the percentage of the total ratings represented by each individual question's weighted average.

![Diagram of survey scorecard history]

**Figure 172: Average ratings detail**

Survey scorecard history
The History view compares the current ratings for the categories and their questions with ratings from the previous three years or four quarters.

Ratings that have declined are highlighted in red and display negative numbers. Ratings that have improved are highlighted in green with positive numbers. Arrow icons beside the values in the Diff column indicate the trend of the current survey against the previous survey.

Point to a category to display a line chart that shows the rating trend for that category. Click a category to view the Survey Category form containing the survey questions.
To calculate the current ratings, the system averages the ratings from the trailing twelve month (TTM) period. The Diff column shows the discrepancy between the current ratings and the previous calendar year’s ratings.
## Service Desk Satisfaction Survey

### Survey Scorecard

<table>
<thead>
<tr>
<th>Overall Rating</th>
<th>Current</th>
<th>Diff</th>
<th>2013</th>
<th>2012</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Desk Satisfaction Survey</td>
<td>5.40</td>
<td>0.00</td>
<td>5.40</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>How satisfied were you with the response time to your incident? (1=not at all, 6=completely)</td>
<td>0.00</td>
<td>-1.02</td>
<td>1.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>How courteous and respectful was the technician who responded? (1=poor, 6=excellent)</td>
<td>0.00</td>
<td>-1.21</td>
<td>1.21</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Please rate the technical competency of the technician serving you (1=poor, 6=excellent)</td>
<td>0.00</td>
<td>-1.02</td>
<td>1.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Was technician able to resolve your issue during the first call?</td>
<td>0.00</td>
<td>-1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>How satisfied are you with your overall service experience? (1=not at all, 6=completely)</td>
<td>0.00</td>
<td>-1.14</td>
<td>1.14</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Figure 174: Survey scorecard history - 3 years
4 quarters

Quarterly surveys compare the average rating for each question and category in the current quarter against the average ratings from the previous four quarters. The **Diff** column shows the discrepancy between the current ratings and the previous quarter's ratings. The column labels count backward, by quarter from the current quarter. For example, if the current quarter is the 3rd quarter of 2013, then the previous quarters appear as 2nd [2013], 1st [2013], 4th [2012], and 3rd [2012]. All four of the previous quarters appear, whether or not there was any data for those quarters.
Figure 175: Survey scorecard history - 4 quarters
Export a scorecard as an image
Role required: admin or survey_admin

Click the menu icon (⋮) and select **Save as PNG** or **Save as JPEG** as the download format.

When the export is complete, select **Download** to save the scorecard image to a storage location.

**Survey Administrator tasks**

Survey administrators create and administer surveys. They know what types of surveys are necessary, and when and how to send a survey to survey readers.
Survey administrators can use all modules in the Survey Management application. The survey_admin role contains the survey_reader and assessment_admin roles.

*Create a survey in the Survey Designer*

When you create a survey, you can create one or more categories and then add questions to each category.

Role required: admin or survey_admin

You can assign a survey to individual users or groups who receive all the questions from all the categories. You can also customize each question and make it dependent on the response to another question. Create a survey using these procedures:

- Create the categories.
- Create questions within each category.
- Configure survey details, such as introductory and closing remarks and time limit.
- Select recipients for the survey.
- Publish the survey to the selected users or groups.

**Survey designer**

Users with the survey_admin role can use the survey designer to create, edit, and distribute surveys.

The survey designer replaces the survey creator. If you are using Internet Explorer version 8 or earlier, the system redirects you to the survey creator.

**Survey designer elements**

The survey designer is accessible from [Survey > Survey Designer](survey_designer). The survey designer contains controls, a header bar, and the design canvas.

**Controls**

To create a question, drag the appropriate data type control from the [Controls](controls) palette and drop it onto the designer canvas.
The assessment engine provides a built-in result calculation feature that converts responses to scored question data types to a score between 0 and 10.
### Table 131: Question data types

<table>
<thead>
<tr>
<th>Data type</th>
<th>Scored</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Yes</td>
<td>Question with a check box or Yes and No choices for user responses.</td>
</tr>
<tr>
<td>Choice</td>
<td>Yes</td>
<td>List of predefined options. For more information, see the definition of the <strong>Choices</strong> field in Create a question in the survey designer on page 549.</td>
</tr>
<tr>
<td>Date</td>
<td>No</td>
<td>Date field.</td>
</tr>
<tr>
<td>Date/Time</td>
<td>No</td>
<td>Date and time field.</td>
</tr>
<tr>
<td>Number</td>
<td>No</td>
<td>Number field with predefined minimum and maximum values. The default is 1-10.</td>
</tr>
<tr>
<td>Percentage</td>
<td>No</td>
<td>Percentage field with a prescribed range.</td>
</tr>
<tr>
<td>Scale</td>
<td>Yes</td>
<td>Predefined Likert scale. Answer options appear as radio buttons.</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>Yes</td>
<td>Selectable number scale. The default is 1-5. Answer options appear as radio buttons.</td>
</tr>
<tr>
<td>String</td>
<td>No</td>
<td>Single or multi-line text field.</td>
</tr>
<tr>
<td>Template</td>
<td>Yes</td>
<td>Choice list of templates that provide a predefined scale of options. For details, see Configure a Template Question.</td>
</tr>
<tr>
<td>Reference</td>
<td>No</td>
<td>Choice list of fields from a specified reference table. This data type does not support reference qualifiers. For example, a user could select a user name if you specify sys_user as the reference table.</td>
</tr>
</tbody>
</table>

### Header bar

The tabs on the header bar display views and a menu of functions.
Figure 177: Survey Designer header bar and menu

Click a tab to change the view on the canvas:

- **Design**: Add and configure the properties of categories and questions. This is the default view.
- **Configuration**: Create introductions and end notes for surveys and select a *signature*.
- **Availability**: Select the recipients for each category in the survey.

Point to the menu icon () to select an option. The list of options depends on the currently open survey.

- **Save**: Save the current survey.
- **Preview**: Display a preview of the survey as it will appear to recipients.
- **Publish**: Distribute the survey to the selected recipients.
- **Save and Publish**: Save and distribute the survey in one step.
- **New Survey**: Open a fresh canvas for a new survey.
- **Load Survey**: View the list of existing surveys.

**Design canvas**

New surveys open in the canvas of the **Design** view. The survey **Name** field appears above first category in the canvas. A blank question field appears in the category container.
Configure a survey in the survey designer
The configuration values you use in this procedure are applied to the entire survey.
Role required: admin or survey_admin
1. Select **Configuration** in the survey designer.
2. Complete the Survey Designer Configuration form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Check box for enabling the distribution of this survey to recipients.</td>
</tr>
<tr>
<td>Anonymize responses</td>
<td>Check box for collecting survey responses anonymously. Recipients are not listed with survey results.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of this configuration or the survey to which it is attached.</td>
</tr>
<tr>
<td>Introduction</td>
<td>Introductory content to display on surveys. You can add a welcome message or background information about the survey.</td>
</tr>
<tr>
<td>Signature</td>
<td>Acknowledgement by a survey recipient of requirements, admonitions, or expectations related to a survey.</td>
</tr>
<tr>
<td>Return URL</td>
<td>Destination address of a web page that is presented to users after they submit a completed survey. When a return URL is configured, the <strong>End note</strong> content does not appear.</td>
</tr>
</tbody>
</table>

**Figure 178: Initial appearance of the Survey Designer canvas**
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End note</td>
<td>Content that is displayed to recipients after they submit a completed survey. You can add a thank you message, follow-up instructions, or other applicable information. End notes are not displayed if a <strong>Return URL</strong> is specified.</td>
</tr>
<tr>
<td>Duration</td>
<td>Amount of time that recipients are given to complete this survey, starting from the time that the survey is generated. The default duration is 14 days.</td>
</tr>
</tbody>
</table>

**Note:** By default, the system runs a script every 30 days to cancel expired survey and assessment instances that are in the **Work in progress** or **Ready to take** states. The schedule item is called **Cancel Expired Assessments**.

Create a category in the survey designer

A category represents a theme for evaluating a specific element of the survey topic and contains questions pertaining to that theme.

Role required: admin or survey_admin

When you create a survey, the system creates a default category, using the name of the survey. You can use this category, modify it, or create additional categories as needed. To have any results, a category must contain scored questions. For simple surveys, one category is usually enough.

Consider creating additional categories to:

- Separate groups of questions into collapsible sections by category on the survey questionnaire.
- Report on category scores, which are calculated based on responses for all questions within each category.

1. Navigate to **Survey > Survey Designer**.
2. Enter the name of the survey in the **Name** field. The system uses this name as the name of the survey and the first category.
3. To configure the category, click in the gear icon in its title bar. The Properties dialog box appears. You can change the name of the category, add a description for it, and enter text in the Details field that introduces or explains the category to recipients.
4. Click the X icon to close the category properties dialog box and save your settings.
5. To add a new category, click the + icon in the title bar of an existing category.

Create a survey category
Survey administrators can create and administer survey categories.
Role required: admin or survey_admin
Only one category is required for each survey, but you can create as many as you like.

1. Navigate to Survey Management > View Surveys.
2. Open a survey definition.
3. In the **Metric Categories** related list, open a category or click **New**. Each survey category is stored as a record on the Metric Category [asmt_metric_category] table with a modified view for survey use.

4. Complete the Survey Category form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the category. The category name appears on questionnaires when:</td>
</tr>
<tr>
<td></td>
<td>• There is more than one category for the survey.</td>
</tr>
<tr>
<td></td>
<td>• There is only one category and its name is different than the survey</td>
</tr>
<tr>
<td></td>
<td>definition name. If you create a survey using the survey creator, the</td>
</tr>
<tr>
<td></td>
<td>category name is the same as the survey definition name.</td>
</tr>
<tr>
<td>Description</td>
<td>Helpful information about the category.</td>
</tr>
<tr>
<td>Related Lists</td>
<td></td>
</tr>
<tr>
<td>Assessment</td>
<td>All survey questions in this category.</td>
</tr>
<tr>
<td>Metrics</td>
<td></td>
</tr>
<tr>
<td>Users</td>
<td>All survey users for this survey.</td>
</tr>
</tbody>
</table>

5. Save the record.

Survey categories
Survey categories provide a way to group questions of a similar theme for a given survey.

There must be at least one survey category per survey definition and every survey question must be associated with a category. When you create a survey, the system generates one category, which all the questions belong to.

For simple surveys, one category is usually enough. Consider creating additional categories if you want to accomplish any of the following tasks.

- Separate groups of questions into collapsible sections by category on the survey questionnaire.
- Report on category scores, which are calculated based on responses for all questions within each category.

Create a question in the survey designer
You can create multiple questions for each category, but each question can only be associated with one category.

Role required: admin or survey_admin

The data type that you select for each question determines how it can be answered by survey recipients.

1. In the Design view, drag a data type icon from the Controls palette and drop it into a category container.
2. To configure the question, click in the gear icon in its title bar. The Properties dialog box appears for the question.
3. Complete the form.
## Table 134: Question property fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Required] Name of the question. Create a concise and easily recognizable name for your question. The system uses this value to identify the question in Assessment Metric lists and in scorecard charts.</td>
</tr>
<tr>
<td>Question</td>
<td>Text to display as the question on surveys. Enter a clear, straightforward question that is easy to understand.</td>
</tr>
<tr>
<td>Type</td>
<td>[Read-only] Data type selected for this question. See the table in Controls for possible data types.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that determines whether this question is available on a survey. If a question is marked inactive, it does not appear on surveys generated after the question becomes inactive.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Check box for requiring users to answer the question. Users cannot submit surveys until they answer all mandatory questions, which are denoted by a red field status indicator.</td>
</tr>
<tr>
<td></td>
<td>This field is available when the question does not have a dependency and the question Controls is not Boolean with a check box option.</td>
</tr>
<tr>
<td>Boolean option</td>
<td>Whether a check box or a Yes/No list appears as the option for the Boolean question.</td>
</tr>
<tr>
<td>String option</td>
<td>Setting for the appearance of a string field in a question. This field is available when the question type is String. The string options are:</td>
</tr>
<tr>
<td></td>
<td>• Single line: Single line text field 40 characters in length that allows strings of any length.</td>
</tr>
<tr>
<td></td>
<td>• String line wide: Full page width text field that allows a single line entry of any length.</td>
</tr>
<tr>
<td></td>
<td>• Multiline: Full page width multiline text field that allows word wrap and returns.</td>
</tr>
<tr>
<td>Min</td>
<td>Lowest positive whole number that users can enter or select to answer the question. This field is available when the question type is Number, Percentage, or Numeric Scale.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Max</td>
<td>Highest positive whole number that users can enter or select to answer the question. This field is available when the question type is Number, Percentage, or Numeric Scale.</td>
</tr>
<tr>
<td>Allow not applicable</td>
<td>Check box for including Not Applicable as an option for this question. Users can select Not Applicable if they do not have sufficient information to respond to a question. User responses of Not Applicable are excluded from results calculations. This field is available when the question does not have a dependency and the question Data type is not Boolean with a check box option.</td>
</tr>
<tr>
<td>Randomize answers</td>
<td>Check box for displaying answer options in a random order whenever the question appears in a survey. Answer preference is sometimes influenced by the order in which options appear, which can result in biased results. Randomizing options can help prevent this bias. Note: Randomizing options for certain questions may make those questions confusing for users. In general, only randomize options that do not follow a logical order.</td>
</tr>
<tr>
<td>Positive indicator</td>
<td>Setting that determines whether smaller or larger numerical values equate to a good score in result calculations. Select Low values if smaller numerical values are better, such as for a question that measures the number of incidents for a vendor. Select High values if larger numerical values are better, such as for a question that measures user satisfaction on a scale of one to five.</td>
</tr>
<tr>
<td>Details</td>
<td>Information about the question that is displayed on the survey. Include details that help users understand how to answer the question.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Choices</td>
<td>Options for a question with a data type of <strong>Choice</strong> or <strong>Scale</strong>. The system automatically adds text and values that you can edit for each option. Click the + icon to add an option, or click the X icon to delete an option. By default, the system arranges options in the order established by their values. To change the order, drag and drop the options. You must have at least two options, and each option must have a unique value. <strong>Note</strong>: Value numbers also contribute to the calculations of survey response scores, which can be used by advanced survey administrators.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependency</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Displayed when</td>
<td>Condition builder that hides or displays the question depending on the answer to another question in the same category. Select an existing question from the list with a data type of <strong>Boolean</strong>, <strong>Choice</strong>, <strong>Scale</strong>, or <strong>Template</strong>. Create the condition that must exist for recipients to see the dependent question, using the <strong>is</strong> or <strong>is one of</strong> operator. The system prevents recursive dependencies between questions. For example, if Question A depends on Question B, Question B cannot depend on Question A.</td>
</tr>
</tbody>
</table>

4. To create any special conditions that must be met before the question appears on the survey, click the Dependency tab. Fill in the fields, as appropriate. Dependent questions appear on the survey when a recipient selects a specific answer or combination of answers to another question in the same category.

5. Select a question in the **Displayed when** field.
The system selects the appropriate operator and displays the possible answers for the selected question.

6. Select the answers that satisfy the condition. Selected answers are indicated by a check mark.

7. Click the X icon to close the question properties dialog box and save your settings.

8. To add a question with the same data type as an existing question in the category, click the + icon in the title bar of the existing question.

9. Drag and drop questions to change their order within a category or move them between categories.

10. To delete a question, click the X icon in its title bar.

Survey question data types
You must choose a **Data type** for each survey question. The data type determines the format of the question and the kind of data that is collected on survey questionnaires.

The following data types are available for surveys:

**Boolean data type**

On questionnaires, users select a check box beside a statement or leave it cleared.

If you select **Boolean**, you must fill in the **Scale definition** field. Select **High** if it is best when users select the check box.

**Choice data type**

On questionnaires, users select a value from a list of choices.

If you select **Choice**, you must fill in the **Scale definition** field and create answer options. Select **High** for the scale definition if the answer option with the largest metric definition **Value** is best.

**Note:** The system sets the **Min** and **Max** fields automatically based on the **Value** settings for the associated metric definitions.

**Date and Date/Time data types**

The **Date** and **Date/Time** data types are very similar.

- **Date:** On questionnaires, users select a date.
Figure 180: Question of the Date data type

- **Date/Time**: On questionnaires, users select a date and time.

Figure 181: Question of the Date/Time data type
**Likert Scale data type**

On questionnaires, users select a multiple choice value from a custom *Likert scale*. Each answer option is represented by a radio button on the scale. A Likert scale question that evaluates an application’s ease of use might have the answer options *Easy, Average, and Difficult*.

![Likert Scale Example](image)

**Figure 182: Question of the Likert Scale data type**

If you select *Likert Scale*, you must fill in the *Scale definition* field and create answer options. Select *High* for the scale definition if the answer option with the largest metric definition *Value* is best.

**Note:** The system sets the *Min* and *Max* fields automatically based on the *Value* settings for the associated metric definitions.

**Number data type**

On questionnaires, users enter a number.

![Number Example](image)

**Figure 183: Question of the Number data type**

If you select *Number*, you must fill in these additional fields:

- **Scale definition**: Select *High* if a larger number is better, such as for a question that measures the number of sales made in a quarter.
- **Min** and **Max**: Enter the smallest and largest numbers users can enter. It is usually best to state the range of acceptable answers in the question text.

**Percentage data type**

On questionnaires, users enter a number.
If you select **Percentage**, you must fill in these additional fields:

- **Scale definition**: Select **High** if a larger percentage is better, such as for a question that measures the percentage of work an agent has completed.
- **Min and Max**: Enter the smallest and largest numbers users can enter. Generally, 0 and 100 are appropriate minimum and maximum values. If you choose values other than 0 and 100, it is usually best to state the range of acceptable answers in the question text.

### Reference data type

On questionnaires, users select a value from a list that is generated from a specified reference table. The response field supports auto-completion.

**Note**: Reference qualifiers are not supported.

For example, a user could select a user name in response to a question if you specify User [sys_user] as the reference table.
Figure 185: Question of the Reference data type

String data type

On questionnaires, users enter text. When you select String, the String option field appears. Select one of the following options to determine how the string field appears on questionnaires:

- Figure 186: Single line
**Template data type**

On questionnaires, users select a value from a predefined series of answer options. To use this data type, a question template must be defined.

If you select **Template**, you must fill in these additional fields:

- **Template**: Select a template.
- **Scale definition**: Select **High** if the answer option with the largest template definition **Value** is best.

**Yes/No data type**

On questionnaires, users select **Yes** or **No** from a list.
Geneva    ServiceNow    ServiceNow Platform

Figure 189: Question of the Yes/No data type

If you select Yes/No, you must fill in the Scale definition field. Select High if Yes is the best answer.

Edit a survey in the survey designer
You can modify surveys using the survey designer.
Role required: admin or survey_admin

You can edit a survey even after it has been distributed, with these results:
• Added questions are available only on surveys that are distributed after this change.
• Changes to existing questions are immediately available to users before the survey is submitted or during the retake period. This includes changes to the answers, such as additional choices or changes to the data type.
• Deleted questions are also deleted from the distributed surveys in users' queues.

1. Navigate to Survey > Survey Designer.
2. Point to the menu icon in the survey header bar, and select Load Survey.
3. Select a survey from the list and modify it as needed.

Configure category weights for a survey
The system calculates results from the weight configured for each category.
Role required: admin or survey_admin

Weights are set to a value of 10 by default but can be changed.
1. Navigate to Survey > View Surveys and select a survey from the list.
2. In the Survey Definition form, select a category from the Metric Categories related list.
3. In the Survey Category form, click the menu icon in the header bar and select Configure > Form Layout.
4. Move the Weight field to the Selected list, and click Save.
5. Edit the default weight value, and update the record.

View a survey instance
A survey instance represents one questionnaire assigned to one user. You might view an instance to verify that survey instances were created, to check the state of a survey instance, or to reassign a survey instance.
Role required: admin or survey_admin

Navigate to Survey Management > Survey Instances and open a survey instance.
Each survey instance is stored as a record on the Assessment Instance [asmt_assessment_instance] table with a modified view for survey use.
### Table 135: Survey Instance form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Automatically generated record number.</td>
</tr>
<tr>
<td>Metric type</td>
<td>Survey definition this survey instance was created from. Survey definitions are stored on the Assessment Metric Type table, which is why the field is called Metric type.</td>
</tr>
<tr>
<td>Due date</td>
<td>Date by which the survey instance should be completed. The system populates the due date based on the Assessment duration of the associated survey definition, which is set to 14 days by default. For example, a survey instance generated March 2nd would be due March 16th by default. The survey due date is not enforced in the base system. If you want to enforce the due date, consider using a workflow or other mechanism to send survey recipients reminders when a survey is overdue.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> By default, the system runs a script every 30 days to cancel expired survey and assessment instances that are in the Work in progress or Ready to take states. The schedule item is called Cancel Expired Assessments.</td>
</tr>
<tr>
<td>State</td>
<td>State of the survey instance. The possible states are Ready to take, In progress, Complete, and Canceled.</td>
</tr>
<tr>
<td>Assigned to</td>
<td>User this survey instance is assigned to. This field becomes read-only when the state is In progress, Complete, or Canceled.</td>
</tr>
<tr>
<td>Expiration date</td>
<td>Date on which the assigned user can receive a new instance of the same survey definition. The system automatically populates the expiration date based on the Schedule period of the associated survey definition.</td>
</tr>
<tr>
<td>Related List</td>
<td></td>
</tr>
</tbody>
</table>
### Field Description

**Assessment Instance Questions**

All instance question records, which store user response values for each question on the survey questionnaire. The following columns are relevant to surveys:

- **Category**: Displays the survey categories the questions belong to.
- **Metric**: Displays the survey questions.

---

**Select recipients for a survey in the survey designer**

You can use this procedure to determine who receives the survey.

**Role required**: admin or survey_admin

1. After a survey is created and configured, click *Availability* to determine who receives the survey, and specify accessibility by selecting one of the following.

   - **Anyone**: Allows any user with an account on the instance to take the survey. For example, you can send users the survey URL via email.
   - **Specific users**: Allows you to choose specific users, groups, or survey groups.

2. Click *Save*.
Designate a survey user

You can designate one survey user at a time from the Survey Definition form.

Role required: admin or survey_admin

1. Navigate to **Survey Management > View Surveys**.
2. Open a survey definition.
   There must be at least one category.
3. In the **Survey Users** related list, click **New** to open the Survey User form.

   ![Survey User form]

4. Select a **User**.
5. Click **Submit** to return to the survey definition.

   To remove survey users, in the **Survey Users** related list, use the action check boxes and choice list.

Survey users and groups

Survey users and survey user groups help survey administrators control who can take a survey.

Survey administrators can restrict a survey so that only specified users can access it unless a survey administrator manually assigns the survey to a different user. Survey user groups provide a way to quickly designate multiple survey users from the survey creator.

Administering survey users

The list of survey users for a survey is visible on the Survey Definition and Survey Category forms. You can add or remove users from the list of survey users at any point. Note that designating a survey user does not automatically generate a survey instance for that user unless both of the following conditions are true:

- The survey definition **Schedule** period is set to **Daily**, **Weekly**, **Monthly**, or **Yearly**. In this case the system assigns a new survey instance to each survey user at the beginning of each schedule period.
- The user has no instances of the survey that are incomplete or that have not yet reached their expiration date.

You can designate survey users from the survey creator, the Survey Definition form, or the Survey Category form.

**Note:** If there are trigger conditions for a survey, do not create survey users. Instead, use the Trigger Conditions form to assign users.

Survey creator

You can designate multiple survey users using the survey creator. In the Survey User Access section, select the **Selected users can take this survey** option, then use the slushbucket to add users.
Alternatively, select the Members of the selected groups can take this survey option, then use the slushbucket to add survey user groups or other groups. The system saves the users as survey users when you save or publish the survey.

Create a survey user group
Survey user groups are groups that have the Type field set to survey and that display only the information most relevant to surveys.

Role required: admin or survey_admin

When you use the survey creator you can select survey user groups or any other groups whose members you want to assign surveys.

Though it is possible to designate members of any group as survey users, you might want to create a survey group for the added convenience of viewing it in the survey User Groups module.

Note: The user_admin role is required to create and edit groups.

1. Navigate to Survey Management > Administration > User Groups.
2. Click New.
3. Fill in the fields on the Group form and add members, as appropriate.

Designate or remove multiple survey users at one time
Use the Survey Category form to designate or remove multiple survey users at a time.

Role required: admin or survey_admin

1. Open a survey definition.
2. In the Metric Categories related list, open a category.
   
   You can choose any category. The system applies survey user changes to all the survey's categories automatically.

3. In the Users related list, click Edit.
4. Use the slushbucket to add or remove survey users.
5. Click Save to return to the survey category.
   
   The changes are also reflected in the Survey Users related list on the survey definition.

Allow recipients to retake a survey
You can configure a survey to allow recipients to resubmit their answers as many times as they like, up to the survey's due date.

Role required: admin or survey_admin

Results are not calculated until the survey's configured duration has elapsed. The card in the user's queue remains visible until the survey's due date, and a button is displayed to allow retakes.

1. Navigate to **Surveys > View Surveys**.
2. Select a survey from the list.
3. In the Survey Definition form, click the menu icon in the header bar and select **Configure > Form Layout** from the context menu.
4. Move the **Allow retake** field to the **Selected** list, and click **Save**.
5. Select the **Allow retake** check box, and update the form.

---

**Publish a survey**

You must publish a survey to enable people to receive and complete survey instances.

Role required: admin or survey_admin

The **State** field on the Survey Definition form indicates whether the survey is in the **Draft** or **Published** state. If you do not publish the survey when you use the survey creator, open a draft survey definition and click **Publish**.

When you publish a survey, the system generates survey instances for any associated survey users. You can **assign** the survey to other users manually.

**Note:** You cannot return a survey to the **Draft** state after it has been published. You do have the option to deactivate a survey by clearing the **Active** check box.

---

**Publish a survey in the survey designer**

You must save changes to a survey before you can publish it to the specified recipients or groups.

Role required: admin or survey_admin

1. To view the survey as the recipients will view it, point to the menu icon and click **Preview**.
2. When you are satisfied with the survey, click either **Save and Publish** or **Publish** to distribute it.

   When you publish a survey, the system sends email notifications to the recipients and to their managers. The system displays a link to the survey on a card in each recipient's My Assessments & Surveys portal.

---

**Customize the appearance of a survey**

You can set properties to customize the color of various elements on the questionnaires.

Role required: admin or survey_admin

**Note:**

- For color properties, enter either an HTML color name or hexadecimal (hex) value. For hex values, the # character is required. Values are not case-sensitive.
- For example, all of the following formats are valid: LightGray, lightgray, #D3D3D3. A preview of the color appears next to the field.

1. Navigate to either of the following modules.
   - **Assessments > Admin > Assessment Properties**
   - **Survey Management > Administration > Properties**
2. On the properties page, edit the properties as needed. Reference the screenshot below to see what parts of assessment questionnaires are controlled by the properties.

3. Click **Save**.

**Note:** You may need to clear the browser’s cache to see updates.

---

### Editing assessment and survey properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
</table>
| `css.assessment.question.header.background.color` | Assessment question header background color | Sets the background color of question headers on assessment and survey questionnaires.  
  • **Default value:** #767676            |
| `css.assessment.caption.background.color`    | Assessment caption background color | Sets the background color of the caption on assessment and survey questionnaires.  
  • **Default value:** #eee               |
| `css.assessment.caption.font.color`          | Assessment caption font color  | Sets the font color of the caption text on assessment and survey questionnaires.  
  • **Default value:** #ffffff            |

---

**Assessment and survey properties**

Assessment administrators and survey administrators can use properties to customize assessments and surveys. Note that customizations you save apply to all assessments and surveys.

**Change the survey question header color**
All of the lists and forms draw the color of their headers from the base color, allowing theming from the CSS properties. One notable exception is the survey, which has its own set CSS. It is possible to change the CSS by adding a property.

1. Enter sys_properties.form into the navigation filter to display a new Property form.
2. Name the new property css.list.row.survey.background.color and enter the hexadecimal value of the desired color into the Value field.
3. Save the property.

The survey should now use the desired color.

Customize a survey definition
Survey administrators can customize survey definitions by adding options.

Role required: admin or survey_admin

1. Navigate to Survey Management > View Surveys.
   Each survey definition is stored as a record on the Assessment Metric Type [asmt_metric_type] table with a modified view for survey use.
2. Open a survey definition.
3. Modify the fields on the Survey Definition form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the survey, which appears on the questionnaire.</td>
</tr>
<tr>
<td>Description</td>
<td>Helpful information about the survey.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box for activating the survey definition. When the Active check box is cleared, new survey instances cannot be generated and users cannot complete existing survey instances. Use the Active check box to deactivate or activate a survey that is published.</td>
</tr>
<tr>
<td>Anonymize responses</td>
<td>Check box for ensuring that all responses for this survey are stored without the submitting users' names. When a user submits a survey questionnaire, the system clears the Assigned to field for the associated survey instance. Additionally, survey responses for anonymous surveys do not contain Assigned to values.</td>
</tr>
<tr>
<td>State</td>
<td>Status of the survey: Draft or Published.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Schedule period</td>
<td>Option that determines how often a user can take the same survey and whether the system generates survey instances on a schedule. For more information, see Schedule periods on page 575.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: If you use a trigger condition for a survey, ensure the schedule period is set to <strong>No Limit</strong>. Trigger conditions use a different method to regulate how often users can receive survey instances.</td>
</tr>
<tr>
<td>Scheduled job</td>
<td>[Admin only] Scheduled job the system creates for this survey definition when the schedule period is a repeating interval. The system sets the scheduled job to run according to the selected schedule period. If you change the schedule period and save the survey definition, the system deletes the old scheduled job and, if you selected a recurring schedule period, creates a new scheduled job. For example, if you change the schedule period from <strong>Daily</strong> to <strong>Weekly</strong> and save the record, the system deletes the daily scheduled job and creates a weekly one set to run a week from the current date. This field is visible to administrators only if the schedule period is <strong>Daily</strong>, <strong>Weekly</strong>, <strong>Monthly</strong>, or <strong>Yearly</strong>.</td>
</tr>
<tr>
<td>Introduction</td>
<td>Introductory content to display on survey questionnaires. Consider adding a company logo, a welcome message, background information about the survey, or instructions.</td>
</tr>
<tr>
<td>End note</td>
<td>Content to display on the screen that appears when someone submits a survey questionnaire. Consider adding a thank you message, follow-up instructions, or other applicable information.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Assessment duration</td>
<td>Amount of time survey recipients have to complete their assigned survey instances, starting from the time the survey instance is generated. The assessment duration sets the Due date for each survey instance. The default duration is 14 days. You may need to customize the form to see this field.</td>
</tr>
<tr>
<td>Note:</td>
<td>By default, the system runs a script every 30 days to cancel expired survey and assessment instances that are in the Work in progress or Ready to take states. The schedule item is called Cancel Expired Assessments.</td>
</tr>
</tbody>
</table>

**Related Links**

<table>
<thead>
<tr>
<th>Enable Public Access</th>
<th>Link that enables/disables the survey as a public survey. No login is required to take a public survey and users or non-users can respond anonymously. For completed public surveys, the Assigned to field value is Guest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove Public Access</td>
<td></td>
</tr>
<tr>
<td>View Responses</td>
<td>Link that opens the list of responses for this survey. This related link is available only if there are results for the survey.</td>
</tr>
<tr>
<td>View Scorecard</td>
<td>Link that opens the scorecard for this survey. This related link is available only if there are results for the survey.</td>
</tr>
<tr>
<td>View Survey URL</td>
<td>Link that opens a dialog box that displays a URL for this survey. This is useful for sharing the URL of a public survey. This related link is available only if the Active check box is selected. Note that the URL does not work until the survey definition is published.</td>
</tr>
</tbody>
</table>

**Related Lists**

<table>
<thead>
<tr>
<th>Metric Categories</th>
<th>All survey categories for this survey.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey Users</td>
<td>All survey users who are authorized to take this survey. If no users are listed, any user can take this survey.</td>
</tr>
</tbody>
</table>
4. Save the record.

Modify a survey definition

You can configure additional options for a survey.

Role required: admin or survey_admin

1. Navigate to Survey Management > View Surveys.
   Each survey definition is stored as a record on the Assessment Metric Type [asmt_metric_type] table with a modified view for survey use.
2. Open a survey definition.
3. Modify the fields on the Survey Definition form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment Instances</td>
<td>All survey instances for this survey. You may need to configure the form to see this related list.</td>
</tr>
<tr>
<td></td>
<td>Note: If you add this related list to the form, use list control to omit the New button. Survey instances must be generated by the system to produce functional surveys.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the survey, which appears on the questionnaire.</td>
</tr>
<tr>
<td>Description</td>
<td>Helpful information about the survey.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box for activating the survey definition. When the Active check box is cleared, new survey instances cannot be generated and users cannot complete existing survey instances. Use the Active check box to deactivate or activate a survey that is published.</td>
</tr>
<tr>
<td>Anonymize responses</td>
<td>Check box for ensuring that all responses for this survey are stored without the submitting users' names. When a user submits a survey, the system clears the Assigned to field for the associated survey instance. Additionally, survey responses for anonymous surveys do not contain Assigned to values.</td>
</tr>
<tr>
<td>State</td>
<td>Status of the survey: Draft or Published.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Signature   | Acknowledgement by a survey recipient of requirements, admonitions, or policies related to the survey. The signature might require the recipient to select a check box or type a full signature to verify having read these assertions. You can display assertions without requiring a signature. Select an existing signature from the list or click **New** to create a new one. The signature form contains these fields:  
  - **Name**: Descriptive name for this signature.  
  - **Signature type**: Type of signature required. The selections are **Check box**, **Full name**, or **Assertion only**. If **Assertion only** is selected, no signature is required to submit the survey.  
  - **Assertion**: Text you want to display to recipients.  
  By default, a property called Require authentication for user signature requires users to authenticate when providing a full name signature. |
| Schedule period | Option that determines how often a user can take the same survey and whether the system generates survey instances on a schedule.  
  **Note**: If you use a trigger condition for a survey, ensure the schedule period is set to No Limit. Trigger conditions use a different method to regulate how often users can receive survey instances. |
<p>| Allow retake | Check box that allows users to modify their answers to a completed survey. Users can resubmit a survey as many times as they want until the due date. After that date, the system removes the survey from the user's My Assessments &amp; Surveys page. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduled job</td>
<td>[Admin only] Scheduled job the system creates for this survey definition when the schedule period is a repeating interval. The system sets the scheduled job to run according to the selected schedule period. If you change the schedule period and save the survey definition, the system deletes the old scheduled job and, if you selected a recurring schedule period, creates a new scheduled job. For example, if you change the schedule period from <strong>Daily</strong> to <strong>Weekly</strong> and save the record, the system deletes the daily scheduled job and creates a weekly one set to run a week from the current date. This field is visible to administrators only if the schedule period is <strong>Daily</strong>, <strong>Weekly</strong>, <strong>Monthly</strong>, or <strong>Yearly</strong>.</td>
</tr>
<tr>
<td>Introduction</td>
<td>Introductory content to display on survey questionnaires. Consider adding a company logo, a welcome message, background information about the survey, or instructions.</td>
</tr>
<tr>
<td>End note</td>
<td>Content to display on the screen that appears when someone submits a survey questionnaire. Consider adding a thank you message, follow-up instructions, or other applicable information.</td>
</tr>
<tr>
<td>Assessment duration</td>
<td>Amount of time survey recipients have to complete their assigned survey instances, starting from the time the survey instance is generated. The assessment duration sets the Due date for each survey instance. The default duration is 14 days. You may need to configure the form to see this field.</td>
</tr>
</tbody>
</table>

**Note:** By default, the system runs a script every 30 days to cancel expired survey and assessment instances that are in the **Work in progress** or **Ready to take** states. The schedule item is called **Cancel Expired Assessments**.

<table>
<thead>
<tr>
<th>Send notifications</th>
<th>Select the check box to send notifications for the survey when it is published. You may need to configure the form to see the field.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Links</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Enable Public Access</td>
<td>Link that enables/disables the survey as a public survey. No login is required to take a public survey and users or non-users can respond anonymously. For completed public surveys, the <strong>Assigned to</strong> field value is Guest.</td>
</tr>
<tr>
<td>Remove Public Access</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** You can use the **View Survey URL** related link to share the URL with users.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Responses</td>
<td>Link that opens the list of <em>responses</em> for this survey. This related link is available only if there are results for the survey.</td>
</tr>
<tr>
<td>View Scorecard</td>
<td>Link that opens the <em>scorecard</em> for this survey. This related link is available only if there are results for the survey.</td>
</tr>
<tr>
<td>View Survey URL</td>
<td>Link that opens a dialog box that displays a <em>URL</em> for this survey. This is useful for sharing the URL of a public survey.</td>
</tr>
<tr>
<td></td>
<td>This related link is available only if the <strong>Active</strong> check box is selected. Note that the URL does not work until the survey definition is published.</td>
</tr>
</tbody>
</table>

**Related Lists**

<table>
<thead>
<tr>
<th>Metric Categories</th>
<th>All <em>survey categories</em> for this survey.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey Users</td>
<td>All <em>survey users</em> who are authorized to take this survey. If no users are listed, any user can take this survey.</td>
</tr>
<tr>
<td>Assessment Instances</td>
<td>All <em>survey instances</em> for this survey. You may need to <strong>configure the form</strong> to see this related list.</td>
</tr>
</tbody>
</table>

**Note:** If you add this related list to the form, use list control to omit the New button. Survey instances must be generated by the system to produce functional surveys.

4. Save the record.

**Schedule periods**

The available schedule periods are **Only Once**, **No Limit**, **Daily**, **Weekly**, **Monthly**, and **Yearly**.

The default schedule period is **No Limit**. When you set the schedule period to anything but **Only Once** or **No Limit**, the system creates a corresponding scheduled job. The scheduled job does the following:

- Ensures that a user can take one survey instance of the same survey per schedule period.
• Generates a new survey instance for each survey user at the beginning of the new schedule period, as long as the survey user does not have an incomplete instance of that survey. A survey instance is incomplete if the state is not Complete.

For example, if you set the schedule period to Monthly and someone attempts to send survey invitations twice in the same day, the system generates survey instances for the survey users the first time only. At the beginning of the next schedule period, the system generates another survey instance for each survey user who completed the previous one.

Schedule periods are enforced by the Expiration date field on the survey instance. As long as the survey instance expiration date has not passed, the assigned user cannot receive a new survey instance. When the system generates a survey instance and the schedule period is anything but No Limit, the Expiration date field is automatically set to the appropriate date. For example, if the schedule period is Weekly, the expiration date is a week after the survey instance is generated.

If you change the schedule period, the scheduled job updates automatically to the correct schedule. However, users who have survey instances for the survey cannot receive new survey instances until their existing survey instances expire, regardless of the new schedule period. Consider the following example: Sal Pindell receives a survey instance when the schedule period is Weekly. The next day, a survey administrator changes the schedule period to No Limit. Sal cannot receive another survey instance until one of the following actions occur:

• Seven days pass from the time Sal's survey instance was generated.
• A survey administrator deletes Sal's survey instance.

After one of these actions occurs, Sal can receive a new survey instance anytime, as long as he has no incomplete instances of the survey.

Note: If you use a trigger condition for a survey, ensure the schedule period is set to No Limit. Trigger conditions use a different method to regulate how often users can receive survey instances.

Survey definitions
A survey definition is the root record on which a survey is built.

The survey designer generates a survey definition automatically when you save or publish the survey. Survey administrators might want to modify the survey definition to configure additional options for the survey or publish the survey when it is ready for distribution.

Survey administrators and survey readers can also send survey invitations directly to users from the survey definition.

Create a survey designer template question
You can create a question that uses choice lists from a template.

Role required: admin or survey_admin

1. Drag the Template data type icon into a category container.
2. Click the gear icon in the question title bar to open the template properties dialog box.
3. Select a predefined scale from the list.
Question entry fields appear for that template.

4. Enter one or more questions that are appropriate for the template.
5. Click the arrow to the right of a question to configure its properties. You must provide a name for each question.
6. Click the back arrow to return to the template properties dialog box.
7. Configure the properties for the remaining questions.
8. Click the X icon to close the template properties dialog box and save your settings.
Create or modify survey questions
You can create and administer survey questions.
Role required: admin or survey_admin

Note: Changes to a survey, such as the addition of questions or the modification of question templates, do not apply to the existing survey instances immediately. However, the changes apply immediately to any new survey instances that you create after saving the changes.

1. Navigate to Survey Management > View Surveys.
2. Open a survey definition.
3. In the Metric Categories related list, open a category.
4. In the Assessment Metrics related list, open an existing question or click New.
   Each survey question is stored as a record on the Assessment Metric [asmt_metric] table with a modified view for survey use.
5. Complete the Survey Question form, as appropriate.
   The specific fields available depend on the selected Data type.

Table 138: Survey Category form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Required] Name of the question. When you create a survey using the survey creator, the question name is the same as the text in the Question field.</td>
</tr>
<tr>
<td>Question</td>
<td>[Required] Text to use for the question, which appears on survey questionnaires.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data type</td>
<td>[Required] Data type of the question. The fields for the response depend on the data type.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot change the data type if another question depends on this question. See <strong>Depends on</strong>.</td>
</tr>
<tr>
<td>Template</td>
<td>Question template to use for the answer options.</td>
</tr>
<tr>
<td></td>
<td>This field is visible and required only if the data type is <strong>Template</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot change the template if another question depends on this question.</td>
</tr>
<tr>
<td>Scale definition</td>
<td>Setting that determines whether smaller or larger numerical values equate to a good score in result calculations. Select <strong>Low</strong> if smaller numerical values are better, such as for a question that measures the number of incidents for a vendor. Select <strong>High</strong> if larger numerical values are better, such as for a question that measures user satisfaction on a scale of one to five.</td>
</tr>
<tr>
<td></td>
<td>This field is visible and required only when certain data types are selected. For more information, see Survey question data types on page 555.</td>
</tr>
<tr>
<td>Randomize answers</td>
<td>Check box that determines whether to present the answer options for this question in a random order each time a user opens the survey. Answer preference is sometimes influenced by the order in which answer options appear, which can result in biased results. Randomizing answer options can help prevent this bias.</td>
</tr>
<tr>
<td></td>
<td>This field is visible only if the data type is <strong>Choice</strong> or <strong>Likert Scale</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Randomizing answer options for certain questions may make those questions confusing for the person answering. In general, only randomize answer options that do not follow a logical order.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Check box that makes the question mandatory (selected) or optional (cleared) on survey questionnaires. Users cannot submit questionnaires until they provide valid responses to all mandatory questions, which are denoted by a red field status indicator. This field is visible only when the Depends on field is empty and the data type is <em>not</em> Checkbox. Questions that depend on other questions and check box questions cannot be mandatory.</td>
</tr>
</tbody>
</table>
| Allow not applicable | Check box that determines whether to include a Not Applicable answer option for this question on survey questionnaires. User responses of Not Applicable are excluded from results calculations.  
This field is visible only if the data type is Choice, Likert Scale, Template, or Yes/No.  |
| Depends on      | Setting used to make this a conditional question, meaning it only appears when users answer another question a certain way. To make a question depend on another question, select an existing question from the list, which displays Checkbox, Choice, Template, and Yes/No questions of the same category as this question. Then, use the Displayed when field to set the conditions that cause this question to appear on surveys. The system prevents the creation of recursive dependencies between questions. For example, if Question A depends on Question B, Question B cannot depend on Question A.  
The triggered survey example demonstrates how to create and use conditional questions. |
<p>| Displayed when  | Answer options for the selected Depends on question which, when chosen on surveys, display this question. This field is visible and required only when the Depends on field is set.  |
| Min             | Smallest numerical value to be used as an answer option for this question. This field is visible and required only if the data type is Choice, Likert Scale, Number, or Percentage.  |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>Largest numerical value to be used as an answer option for this question. This field is visible and required only if the data type is Choice, Likert Scale, Number, or Percentage.</td>
</tr>
<tr>
<td>String option</td>
<td>Selection that determines what kind of response text box appears for this question on survey questionnaires. This field is visible and required only if the data type is String.</td>
</tr>
<tr>
<td>Related Lists</td>
<td></td>
</tr>
<tr>
<td>Assessment Metric Definitions</td>
<td>Answer options for this question. This related list is available only if the Data type is Choice or Likert Scale.</td>
</tr>
</tbody>
</table>

6. Save the record.

Be sure to create answer options if you select the Choice or Likert Scale data type.

---

Survey questions

Survey questions appear on survey questionnaires for the associated survey definition.

The survey creator generates questions and answer options automatically. However, it provides only the basic configuration options for questions, like the question text and the data type. You might want to create additional questions or set advanced configuration options for the questions, including making a question appear conditionally or making a question mandatory. You can also use question templates to define reusable sets of answer options.
Survey questions are available from **Survey Management > Questions**. The list displays information about each question, such as the associated survey definition, which appears in the **Type** column, and the data type. Survey administrators can modify these questions.

**Survey question data types**

You must choose a **Data type** for each survey question. The data type determines the format of the question and the kind of data that is collected on survey questionnaires.

The following data types are available for surveys:

**Boolean data type**

On questionnaires, users select a check box beside a statement or leave it cleared.

If you select **Boolean**, you must fill in the **Scale definition** field. Select **High** if it is best when users select the check box.

**Choice data type**

On questionnaires, users select a value from a list of choices.

![Choice data type example](image)

**Figure 190: Question of the Choice data type**

If you select **Choice**, you must fill in the **Scale definition** field and create answer options. Select **High** for the scale definition if the answer option with the largest metric definition **Value** is best.

**Note**: The system sets the **Min** and **Max** fields automatically based on the **Value** settings for the associated metric definitions.

**Date and Date/Time data types**

The **Date** and **Date/Time** data types are very similar.

- **Date**: On questionnaires, users select a date.
Figure 191: Question of the Date data type

- **Date/Time:** On questionnaires, users select a date and time.

Figure 192: Question of the Date/Time data type
Likert Scale data type

On questionnaires, users select a multiple choice value from a custom *Likert scale*. Each answer option is represented by a radio button on the scale. A Likert scale question that evaluates an application's ease of use might have the answer options *Easy*, *Average*, and *Difficult*.

![Likert Scale Example](image)

Figure 193: Question of the Likert Scale data type

If you select *Likert Scale*, you must fill in the *Scale definition* field and create answer options. Select *High* for the scale definition if the answer option with the largest metric definition *Value* is best.

**Note:** The system sets the *Min* and *Max* fields automatically based on the *Value* settings for the associated metric definitions.

Number data type

On questionnaires, users enter a number.

![Number Example](image)

Figure 194: Question of the Number data type

If you select *Number*, you must fill in these additional fields:

- **Scale definition:** Select *High* if a larger number is better, such as for a question that measures the number of sales made in a quarter.
- **Min** and **Max:** Enter the smallest and largest numbers users can enter. It is usually best to state the range of acceptable answers in the question text.

Percentage data type

On questionnaires, users enter a number.
Figure 195: Question of the Percentage data type

If you select **Percentage**, you must fill in these additional fields:

- **Scale definition**: Select **High** if a larger percentage is better, such as for a question that measures the percentage of work an agent has completed.
- **Min and Max**: Enter the smallest and largest numbers users can enter. Generally 0 and 100 are appropriate minimum and maximum values. If you choose values other than 0 and 100, it is usually best to state the range of acceptable answers in the question text.

**Reference data type**

On questionnaires, users select a value from a list that is generated from a specified reference table. The response field supports auto-completion.

**Note**: Reference qualifiers are not supported.

For example, a user could select a user name in response to a question if you specify User [sys_user] as the reference table.
String data type

On questionnaires, users enter text. When you select **String**, the **String option** field appears. Select one of the following options to determine how the string field appears on questionnaires:

- Figure 197: Single line
Template data type

On questionnaires, users select a value from a predefined series of answer options. To use this data type, a question template must be defined.

If you select Template, you must fill in these additional fields:

- **Template**: Select a template.
- **Scale definition**: Select **High** if the answer option with the largest template definition **Value** is best.

Yes/No data type

On questionnaires, users select **Yes** or **No** from a list.
If you select Yes/No, you must fill in the Scale definition field. Select High if Yes is the best answer.

Create a survey question template
You can create and administer question templates.

Role required: admin or survey_admin

| Note: Changes to a survey, such as the modification of question templates, apply to existing survey instances immediately. |

1. Navigate to Survey Management > Templates.
   Each template is stored as a record on the Assessment Metric Template [asmt_template] table.

2. Click New.

3. Fill in the Name field and save the record.

4. In the Assessment Template Definitions related list, click New.
   Create a template definition for each answer option you want to appear on a question.

5. Fill in the following fields:
   - Display: Enter the text to appear as the answer option.
   - Value: Enter a numeric value, greater than or equal to zero, to which the answer option equates.
     Values are used in results calculations. When you view questions that use templates, answer options appear in order from smallest to largest Value. Each template definition for a given template must have a unique Value.

6. Click Update.
Note: Any templates you create are available for use with both surveys and assessments.

Survey question template
Question templates define reusable sets of answer options for survey questions.

Question templates define reusable rating scales for answering questions, where each answer option on the scale is a template definition. For example, the template named Satisfaction represents a satisfaction scale and contains the following template definitions: Very Satisfied, Satisfied, Neutral, Dissatisfied, and Very Dissatisfied.

Figure 201: Template definition

Templates are available for survey questions that have Data type set to Template. The following question templates are available by default.

Table 139: Default question templates

<table>
<thead>
<tr>
<th>Name</th>
<th>Template Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount</td>
<td>None, Few or little, Average amount, Many, Quite a lot</td>
</tr>
<tr>
<td>Complexity</td>
<td>Very Complex, Complex, Moderate, Simple, Very Simple</td>
</tr>
<tr>
<td>Frequency</td>
<td>Never, Seldom, Sometimes, Most of the time, All of the time</td>
</tr>
<tr>
<td>Name</td>
<td>Template Definitions</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>Likert 5</td>
<td>Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree</td>
</tr>
<tr>
<td>Quality</td>
<td>Very Poor, Poor, Average, Good, Very Good</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>Very Dissatisfied, Dissatisfied, Neutral, Satisfied, Very Satisfied</td>
</tr>
<tr>
<td>Size</td>
<td>Very Small, Small, Average, Large, Very Large</td>
</tr>
</tbody>
</table>

Update min and max values to match templates

If you use survey result calculation data, ensure that the Min and Max values for a question that uses a template are equal to the smallest and largest template definition Value.

Role required: admin or survey_admin

The system sets the Min and Max fields based on the template definition values when you create a question of the Template data type. The system does not, however, update the fields for existing questions if you add a new template definition to a template or if you update the Value of an existing template definition. If the new Value is less than the minimum value or greater than the maximum value of any questions that use the template, update the questions accordingly.

1. Navigate to Survey Management > Questions.
2. Configure the list to show the Min and Max columns.
3. Add the following list filter condition: [Template] [is] [<select the template you updated>]
4. Ensure the Min and Max values match the smallest and largest template definition Value for the selected template.
   If the values do not match, edit the Min and Max directly from the list.

   **Note:** When the data type is Template, a UI policy prevents the editing of Min and Max from the form.

Create or modify answer options

You must create answer options, called metric definitions, for survey questions that have Data type set to Choice or Likert Scale.

Role required: admin or survey_admin

   **Note:** Changes to a survey, such as the addition or modification of answer options, apply to existing survey instances immediately.

1. Open a choice or Likert scale survey question.
2. In the Assessment Metric Definitions related list, open a metric definition or click New.
   Each answer option is stored as a record on the Assessment Metric Definition [asmt_metric_definition] table.
3. Complete the Assessment Metric Definition form.

   **Table 140: Assessment Metric Definition fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>[Required] Text to appear as the answer option.</td>
</tr>
</tbody>
</table>
4. Click **Submit**.

On survey questionnaires, the answer options for a question appear in order from smallest to largest **Value**. For example, consider the survey question **How do you feel?** with the answer options **Good**, **Neutral**, and **Bad**. The following table shows the answer option order based on the **Value**.

**Table 141: Answer option order based on value**

<table>
<thead>
<tr>
<th>Answer Option</th>
<th>Value</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>1</td>
<td>First</td>
</tr>
<tr>
<td>Neutral</td>
<td>3</td>
<td>Second</td>
</tr>
<tr>
<td>Bad</td>
<td>5</td>
<td>Third</td>
</tr>
</tbody>
</table>

If you use survey result calculations, ensure the question **Scale definition** is set appropriately based on the answer options. In the previous example, if you want the answer option **Good** to earn the highest score, the scale definition should be **Low** because **Good** has the smallest value.

**Change the order of survey questions**

You can easily reorder survey questions at the category level or the question level.

**Role required:** admin or survey_admin

You can change the order in which the questions in one category appear relative to those in other categories for the same survey definition. You may want to change the order of questions if you add a new question manually after creating other questions when you use the survey creator.

When you create questions using the survey creator, the system sets the **Order** field for the first question to **101**, the second to **102**, and so on. When you create a new question outside of the survey creator, the **Order** is set to **100** by default, which means it appears before all questions generated by the survey creator.

1. Navigate to **Survey Management > View Surveys** and open the appropriate survey definition.
2. In the **Metric Categories** related list, edit the values in the **Order** column.
3. To change the order of questions within a category, complete the following steps.
   a) Open the **Assessment Metrics** related list.
   b) Edit the values in the **Order** column.
4. Click **Update**.

*Configure a trigger condition for a survey*

You can configure trigger conditions to specify when to send a particular survey and the persons to send it to.

Role required: admin or survey_admin

1. Navigate to **Survey Management > Trigger Conditions**.
2. Click **New**.

   **Note:** Do not specify particular users for a triggered survey because only the specified users will be able to take the survey.

3. Complete the form, as appropriate.

### Table 142: Trigger condition fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment</td>
<td>Survey to send.</td>
</tr>
<tr>
<td>Table</td>
<td>Table to run the trigger condition on. You can select only tables in the current application scope. For example, to send a survey whenever an incident closes, select the Incident [incident] table.</td>
</tr>
<tr>
<td>User field</td>
<td>Field that stores the users you want to send the survey. You can select any field, on the selected table or on a referenced table, that references the User [sys_user] table. Use the tree picker to select a field.</td>
</tr>
<tr>
<td>Repeat interval</td>
<td>Minimum period that must pass before the trigger condition can resend the survey to the same user. For example, assume the repeat interval is set to 30 days. Even if the same user qualifies for multiple surveys from this trigger condition, the system can send her only one survey every 30 days.</td>
</tr>
</tbody>
</table>

**Note:** Ensure that the Schedule period of the selected survey definition is set to **No Limit**. If the schedule period is set to a different value, it may prevent the trigger condition from sending surveys as expected.

<p>| Active      | Check box that determines whether this trigger condition is active (selected). |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business rule</td>
<td>[Admin only] Business rule the system creates to monitor the selected table. When the condition is met, the business rule sends the survey to the correct user. No configuration is necessary for this business rule.</td>
</tr>
<tr>
<td>Trigger randomly</td>
<td>Check box that determines whether to send the survey to the appropriate user every time the condition is met (cleared) or only a percentage of the time (selected).</td>
</tr>
<tr>
<td>Probability (%)</td>
<td>Approximate probability that the survey is sent each time the condition is met. For example, if the probability is set to 50, the system sends the survey approximately 50% of the time the conditions are met, assuming there are no repeat interval restrictions. This field is visible and required only when Trigger randomly is selected.</td>
</tr>
<tr>
<td>Related Field 1-4</td>
<td>Field that contains a value you want to store for reporting purposes. You can pick any reference field on the selected table. When the trigger condition generates a survey instance, the system stores the value from the triggering record. Specify up to four fields. For example, if you select the Incident table, you might select Assigned to and Problem as related fields. The system stores the assigned user and problem associated with the incident as Related record 1 and Related record 2 in the survey instance record. To view the fields, configure the form for any survey instance.</td>
</tr>
<tr>
<td>Description</td>
<td>Summary information to identify the trigger condition.</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition builder that defines the criteria that must be true to send the survey. For example, to send a survey whenever an incident closes, create the condition [State] [is] [Closed].</td>
</tr>
</tbody>
</table>
You can also create a trigger condition directly from the survey creator. If you select **Certain users can take this survey, based on conditions** in the Survey User Access section, the Trigger Condition form opens automatically when you click **Save as Draft** or **Publish**.

---

**Survey trigger conditions**

Trigger conditions specify when to send a particular survey and the persons to send it to.

Survey administrators can use trigger conditions to configure the system to generate a survey instance each time a specified action occurs on a specified table, for example, when an incident or change request closes. The system sends the survey to users that are related to the triggering record, for example, incident callers or change request assignees. You can choose to send a survey every time the condition is met or can set a probability for the system to send a survey at random when the condition is met.

Trigger conditions are ideal for sending transactional surveys. Transactional surveys generally measure satisfaction with a recent experience, such as closing an incident or purchasing an item.

---

**Note:** Trigger conditions are comparable to survey conditions in legacy surveys. If you migrate a legacy survey that has survey conditions, ensure they are deactivated before you recreate them as trigger conditions.

---

**Trigger condition example**

A trigger condition is configured as follows:

- **Assessment:** Service Desk Satisfaction Survey
- **Table:** Incident [incident]
- **User field:** Caller
- **Repeat interval:** 30 days
- **Active:** true
- **Trigger randomly:** false
- **Related Field 1**: Assigned to
- **Related Field 2**: Problem
- **Condition**: [State] [is] [Closed] [or] [State] [is] [Resolved]

Fannie Steese is the caller on an incident that is assigned to Boris Catino, a service desk technician. Boris creates PRB010101 based on Fannie's complaint and closes the incident. The system creates a survey instance assigned to Fannie so she can rate her satisfaction with the incident experience.

Because two related fields were selected as part of the trigger condition, the survey instance stores the following information from the incident:

- Related record 1: User: Boris Catino
- Related record 2: Problem: PRB010101

Note that even though the trigger condition is set to be triggered each time that the conditions are met, the Repeat interval setting ensures that Fannie does not receive another survey another of her incidents closes within 30 days of the first incident.

**Send survey invitations to users**

You can send survey invitations using the Assign Survey or Send Invitations buttons on the Survey Definition form.

Role required: admin or survey_admin

Use the Send Invitations button to immediately assign survey instances to each survey user that is listed in the survey definition. Use the Assign Survey button to assign a survey instance to any one user at a time.

Each of the buttons generates survey instances assigned to the appropriate users. In addition, if the instance is configured to send email, the system generates survey notifications. For either button to be available, the survey definition must meet all the following conditions:

- **Active** check box is selected.
- **State** is Published.
- Associated with at least one question.

**Note**: You (or a trigger) can send more than one instance of a survey to a user at any time.

Navigate to Application > Module.

**Assign a survey to a user**

You can assign more than one instance of a survey to a user at any time.

Role required: admin or survey_admin

1. Navigate to Survey > View Surveys.
2. Open a survey definition.
3. Click Assign Survey.
4. In the dialog box, select a user and click OK.
   - The system creates a survey instance assigned to the user, assuming the user is eligible to receive a new survey instance. When you use the Assign Survey button, the selected user is not saved as a survey user.

**Send invitations to survey users**

Use the Send Invitations button to immediately assign survey instances to each survey user who is listed in the survey definition.
Role required: admin or survey_admin

1. Navigate to Survey > View Surveys.
2. Open a survey definition that is associated with at least one survey user.
3. Click Send Invitations.

   The system creates survey instances assigned to each survey user who is eligible to receive a new survey instance.

Share a survey
There are several ways to make surveys available to users: send users survey invitations, share a survey URL that opens the survey directly, or create a module that opens a survey.

Role required: admin or survey_admin

Sharing survey URLs
You can distribute a survey by giving survey users a survey URL.

Survey administrators can obtain one of the following types of URLs.

- A general URL for a survey, which users can open to take the survey questionnaire.
  
  When someone opens the URL, first the system ensures the person is logged in. Then the system searches for an instance of the associated survey that is assigned to the logged-in user.

  If a survey instance meets these criteria, the appropriate action occurs based on the State of the survey instance (see table). If there are multiple active survey instances (Ready to take or In progress), the instance with the earliest due date opens. If no survey instance meets the criteria, the system creates a new instance for the survey, assigns it to the user, and opens the survey questionnaire.

- A URL for a specific survey instance, which the assigned user can open to take the survey questionnaire.
  
  When someone opens the URL, the system ensures the person is logged in as the user the survey instance is assigned to. Only the assigned user can access the survey questionnaire. The appropriate action occurs, depending on the State of the survey instance (see table).

Table 143: Survey states

<table>
<thead>
<tr>
<th>Survey instance state</th>
<th>Action upon opening URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready to take: User has not saved any responses.</td>
<td>The survey questionnaire appears for the user to begin. The user can save or submit responses.</td>
</tr>
<tr>
<td>In progress: User has saved at least one response.</td>
<td>The survey questionnaire appears for the user to continue. Any previously saved responses are displayed. The user can save or submit responses.</td>
</tr>
</tbody>
</table>
| Complete: User has submitted all required responses. | If the schedule period is No Limit or if the period restriction has expired, the survey questionnaire for a new survey instance appears for the user to begin.  
If the schedule period restricts the number of times a user can take the same survey, and the period restriction has not expired since the user last completed survey, an error message appears. |
If someone opens a URL for a survey that is deactivated or that has not been published, an error message appears.

Obtain and distribute a general survey URL
You can distribute a general survey URL to allow any eligible user to open a survey questionnaire.

Role required: admin or survey_admin

The general URL is available on the Survey Definition form and on the survey creator.

1. Navigate to Survey Management > View Surveys.
2. Open a survey definition from the Assessment Metric Types list.
3. Under Related Links, click View Survey URL.
   This related link is visible only if the survey definition is Active.
4. In the dialog box that appears, select and copy the URL.
5. Click OK or Cancel to close the dialog box.
6. Distribute the URL to users as needed.

   When a survey user clicks the general survey URL that you provide, the system creates a survey instance for the survey user (if the user does not have an instance in the Ready to take state). The system is configured by default to send an email notification when a survey instance is generated for a user. As a result, the survey user receives a second notification in addition to the notification that you sent with the general survey URL. This might confuse the survey user. To avoid this issue, you can deactivate auto-notification.

Deactivate auto-notification for surveys
When you send a survey notification with a general survey URL to a survey user, the user might receive a second system-generated notification. To avoid this issue, you can deactivate auto-notification.

Role required: admin or survey_admin

1. Navigate to System Notification > Email > Notifications.
2. In the list of Notifications, click the Survey User Invite notification.
3. Customize the form to add the Send to event creator check box.
4. The Send to event creator check box is selected by default. Clear the check box to cause the system to not send the auto-notification messages to survey users.

Obtain a survey instance URL
You can distribute a survey instance URL to allow the assigned user to open a survey questionnaire.

Role required: admin or survey_admin

The survey instance URL is available on the survey instance record.

1. Navigate to Survey Management > Survey Instances.
2. Open the instance for the survey.
3. Under Related Links, click View Instance URL.
   The related link is available only when the survey instance State is Ready to take or In progress and the associated survey definition is Active.
4. In the dialog box that appears, select and copy the URL.
5. Click OK or Cancel to close the dialog box.
6. Distribute the URL to the assigned user as needed.

Create a survey module
You can create a module that opens a survey.

Role required: admin or survey_admin
When a user clicks a survey module, the system creates a new survey instance, opens an existing one, or displays an error message, depending on configuration options for the survey and other factors.

1. Perform the appropriate action for your version of the UI:

| UI16 | 1. Navigate to **System Definition > Application Menus**.  
2. Open the application menu to which you want to add the survey module. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UI15 or UI11</td>
<td>Right-click the application menu you want to add the module to and select <strong>Edit Application Menu</strong>.</td>
</tr>
</tbody>
</table>

2. In the Modules related list, click **New**.

3. Fill in the following fields:
   - **Link type**: Assessment
     Do not select **Survey**, which is used for legacy surveys only.
   - **Assessment**: Select the survey you want the module to open.

4. Complete the form, as appropriate.

Exporting and importing surveys

Export a survey to import it to a different ServiceNow instance.

The system exports a single XML file that contains a survey definition [asmt_metric_type] and the records that it is associated with:

- Assessment Metric Template [asmt_template]
- Assessment Template Definition [asmt_template_definition]
- Metric definitions [asmt_metric_definition] (survey question answer options)
- Scheduled Jobs associated with the Survey [sys_trigger]
- Survey categories [asmt_metric_category]
- Survey questions [asmt_metric]
- Survey users [asmt_m2m_category_user]
- Trigger Condition [asmt_condition]

Export a survey

You can export a survey.

Role required: admin, survey_reader, or survey_admin

1. Navigate to **Survey Management > View Surveys**.
2. Right-click the name of a record to show the context menu.
3. Select **Export Assessment**.
4. If applicable, follow the prompt in your browser to save the XML file.

Export a survey scorecard as an image

You can export a scorecard as an image in either the PNG or JPEG format.

Role required: admin, survey_reader, or survey_admin

1. Open the scorecard to the view you want to export.
2. Click the menu icon in the scorecard and select **Save as PNG** or **Save as JPEG** as the download format.
3. When the export is complete, select **Download** to save the scorecard image to a storage location.
Import a survey
You can import a survey that has been exported as an XML file.

Role required: admin or survey_admin

Ensure that the target instance has assessments enabled. Then import the records as XML data. Note that the exported XML file does not contain result data.

Migrate a legacy survey
Migrate a legacy survey and its related records to take advantage of a more powerful feature set.

Role required: admin or survey_admin
To avoid timing out for very large surveys, you can disable the transaction quota. See *Transaction quotas* on page 2032.

1. Navigate to **Survey Management > Legacy Surveys > Masters**.
   The list of legacy survey masters appears. By default, the list shows only survey masters that have not been migrated. To show all survey masters, remove the **Assessment is empty** breadcrumb.

2. Open a survey.

3. Under **Related Links**, click **Migrate to Assessment**.
   A dialog box appears, which describes what happens when you migrate the survey. Note that certain types of survey questions cannot be migrated.

4. Click **OK**.
   The system generates records on assessment tables based on the survey master, eligible questions and choices, results, and other related survey components. The original survey components are unaffected.

   System messages may appear at the top of the Survey form to:
   - Notify you of questions that could not be migrated.
   - Advise you to review any migrated questions with the **Multiple Choice** type.

5. Optional: View the new assessment survey by clicking the reference icon (🔗) beside the **Assessment** field, which appears on the Survey form when the legacy survey has been migrated.

6. Optional: Remove survey users from the assessment survey if you want all users to be able to take it. When you migrate a survey, any users who have taken it are automatically associated with the assessment survey and become survey users. When there are survey users associated with an assessment survey, only those users can take it.

7. Deactivate any survey conditions associated with the survey by completing the following steps.
   a) Navigate to **Survey Management > Legacy Administration > Survey Conditions**.
   b) In the **Active** column, ensure the value is false for any survey conditions that reference the migrated survey.
   c) Navigate to **Survey Management > Administration > Trigger Conditions** to create new trigger conditions.

The **Migrate to Assessment** related link on the legacy survey record becomes unavailable after the migration. However, if you delete the record referenced in the **Assessment** field, the related link reappears and you can migrate the legacy survey again.

Migrate legacy surveys to the current survey system
Users with the survey_admin role can migrate legacy survey data to create copies of legacy surveys and their related records in assessment tables.

The following legacy survey components are migrated:
- Survey masters
- Supported survey questions and question choices
- Survey instances
- Survey responses

Legacy survey conditions are not migrated and must be recreated as trigger conditions.

**Note:** Survey wizards cannot be migrated.

Survey management, built on the assessment engine, is available as an alternative to legacy surveys.

Survey question migration
Before you migrate a legacy survey, understand that some legacy survey questions cannot be migrated due to incompatible question types.

Legacy survey questions are migrated from the Survey Question [survey_question_new] table to the Assessment Metric [asmt_metric] table. In legacy surveys, the Type field on the Survey Question table determines how the question renders on the survey questionnaire. In assessment surveys, the Data type field on the Assessment Metric table serves a similar purpose. Certain legacy survey types are not supported in assessment surveys.

The following table shows the conversion path for each legacy survey type to an assessment data type, if there is one.

<table>
<thead>
<tr>
<th>Legacy survey type</th>
<th>Assessment survey data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break</td>
<td>Not available</td>
</tr>
<tr>
<td>CheckBox</td>
<td>Checkbox</td>
</tr>
<tr>
<td>Date</td>
<td>Date</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Date/Time</td>
</tr>
<tr>
<td>HTML</td>
<td>Not available</td>
</tr>
<tr>
<td>Label</td>
<td>Not available</td>
</tr>
<tr>
<td>List Collector</td>
<td>Not available</td>
</tr>
<tr>
<td>Lookup Multiple Choice</td>
<td>Not available</td>
</tr>
<tr>
<td>Lookup Select Box</td>
<td>Not available</td>
</tr>
<tr>
<td>Macro</td>
<td>Not available</td>
</tr>
<tr>
<td>Macro with Label</td>
<td>Not available</td>
</tr>
<tr>
<td>Multi Line Text</td>
<td>String (String option set to Multiline)</td>
</tr>
<tr>
<td>Multiple Choice</td>
<td>Likert Scale</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>Number</td>
</tr>
<tr>
<td>Reference</td>
<td>Not available</td>
</tr>
<tr>
<td>Select Box</td>
<td>Choice</td>
</tr>
<tr>
<td>Single Line Text</td>
<td>String (String option set to Single line)</td>
</tr>
<tr>
<td>UI Page</td>
<td>Not available</td>
</tr>
<tr>
<td>Wide Single Line Text</td>
<td>String (String option set to Single line wide)</td>
</tr>
<tr>
<td>Yes / No</td>
<td>Yes/No</td>
</tr>
</tbody>
</table>

Migrated components
When you migrate a survey, the system maps records from survey tables to assessment tables.

To create a functional survey on the assessment framework, the system converts survey records to the most logical equivalent assessment survey records. This may mean multiple assessment survey records represent one legacy survey record.
Review migrated questions

If you want to maintain accurate result calculations, you may need to make minor adjustments to some of the migrated survey records to ensure results are calculated correctly.

**Note:** The assessment engine provides a built-in result calculation feature that converts each survey response to a score between 0 and 10. The configuration required to maintain accurate result calculations is advanced and is not recommended for basic survey implementations. If you do not plan to use result calculation data, ignore the information in this section.

For each legacy question migrated, the system creates a survey question record on the Assessment Metric [asmt_metric] table. For legacy questions of the **Multiple Choice** and **Select Box** types, the system also creates a metric definition record, on the Assessment Metric Definition [asmt_metric_definition] table, for each legacy question choice.

Metric definitions and survey question choices are comparable and have similar fields:

<table>
<thead>
<tr>
<th>Survey question choice field</th>
<th>Related metric definition field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Display</td>
</tr>
<tr>
<td>Value (can be any string value)</td>
<td>Value (must be a numerical value)</td>
</tr>
<tr>
<td>Order</td>
<td>Value (metric definition with smallest Value is first)</td>
</tr>
</tbody>
</table>

When the system migrates legacy survey question choices, it uses the legacy Order to set each metric definition Value. For the legacy question choice with the smallest Order, the corresponding metric definition Value is set to 1. For the legacy question choice with the next smallest Order, the metric definition Value is 2, and so on.

The Scale definition field on the migrated Survey Question form determines whether smaller or bigger metric definition values equate to a good score in survey result calculations. By default, the scale definition is set to **0**, meaning that a lower metric value indicates a better result.
is set to **High**, meaning bigger values are good. When you migrate a legacy survey, check that the default scale definition makes sense for each question.

For example, the following tables depict a sample migrated question and the metric definitions automatically created for it. Recall that the system uses the order of the legacy survey question choices to set the metric definition value. **Excellent** has the lowest **Order** value, so when the system creates a metric definition for this question choice, the **Value** is set to 1. In this case the default scale definition value, **High**, does not make sense, as the system will calculate the worst scores for responses of **Excellent**.

### Table 144: Legacy question: Please rate the overall quality of your service.

<table>
<thead>
<tr>
<th>Question choice text</th>
<th>Question choice value</th>
<th>Question choice order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>A</td>
<td>100</td>
</tr>
<tr>
<td>Good</td>
<td>B</td>
<td>200</td>
</tr>
<tr>
<td>Fair</td>
<td>C</td>
<td>300</td>
</tr>
<tr>
<td>Poor</td>
<td>D</td>
<td>400</td>
</tr>
</tbody>
</table>

### Table 145: Migrated question: Please rate the overall quality of your service.

<table>
<thead>
<tr>
<th>Metric definition display</th>
<th>Metric definition value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>1</td>
</tr>
<tr>
<td>Good</td>
<td>2</td>
</tr>
<tr>
<td>Fair</td>
<td>3</td>
</tr>
<tr>
<td>Poor</td>
<td>4</td>
</tr>
</tbody>
</table>

If there are no **Order** values for the legacy survey question choices, the system sets each corresponding metric definition **Value** based on the alphanumeric order of the legacy **Text** value:

### Table 146: Migrated question: Please rate the overall quality of your service.

<table>
<thead>
<tr>
<th>Metric definition display</th>
<th>Metric definition value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>1</td>
</tr>
<tr>
<td>Good</td>
<td>2</td>
</tr>
<tr>
<td>Fair</td>
<td>3</td>
</tr>
<tr>
<td>Poor</td>
<td>4</td>
</tr>
</tbody>
</table>

You may also need to change the **Scale definition** setting for other migrated questions. For details, see the scale definition recommendations for each survey question data type.

**Take a survey**

Surveys that are assigned to you, and that are not complete, appear in your assessment and survey queue.

Role required: survey_reader, survey_admin, or admin
Each assessment or survey appears as a card in the queue. The card contains helpful information, including the survey name, state, and due date, and a button to launch the questionnaire. Overdue surveys are clearly marked with a red icon and red due date. You must answer every mandatory question, indicated by a red bar, before you can submit the survey. If you start to take a survey but cannot complete it, save your responses and return to it later. When you have answered all the questions and are satisfied with the responses, submit the survey.

1. Navigate to **Self-Service > My Assessments & Surveys**.

   **Note:** Users with the assessment_admin role, including survey administrators, can display other users’ assessments and surveys in addition to their own. Use the **Show all** and **Show assigned to me** related links at the bottom of the queue to show and hide assessments and surveys. Click a card assigned to another user to open the associated metric type or survey definition.

2. Click **Take Survey** on a survey card to open the questionnaire.

   If there is more than one survey category, you can click the collapse or expand icon to hide or show the questions in the category.

3. Answer each question to the best of your ability.

   If you are unsure of how to respond to a question or if a question does not apply to a particular record, select **Not Applicable**, if available.

4. Complete one of the following steps.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Save your responses without submitting them</strong></td>
<td>Click <strong>Save</strong>. You can close the questionnaire and access it later from your queue.</td>
</tr>
<tr>
<td><strong>Submit the survey after answering all questions</strong></td>
<td>Click <strong>Submit</strong>. You cannot return to the questionnaire after submitting.</td>
</tr>
</tbody>
</table>

   **Note:** The system does not save the survey if there are invalid responses, such as a letter in a date field. You must enter valid responses or remove invalid responses before you can save the survey.

**Survey questionnaires**

All surveys that are assigned to you appear in your personal assessment and survey queue.

No special role is required to complete a survey but you must be logged in and the survey must be assigned to you. When you submit a survey, the system stores your responses.

**Survey notification**

If the system is configured to send email, it sends you a notification when a survey is assigned to you. The message contains a link to the survey and instructions for viewing the assessment and survey queue.
Quizzes

Quizzes are questionnaires you can assign to one or more users to assess their knowledge of any subject. The quiz functionality is built on the assessment engine and provides many of the same features as assessments and surveys.

Each question is scored, and the overall score indicates the percentage of questions the user answered correctly. A quiz may have categories of questions that are assigned only to some users. You can assign weighting values to individual questions or categories of questions that make them more or less important when calculating the overall score. Quizzes require activation by a system administrator.

- An administrator can create a quiz for any purpose and assign it to a single user or multiple users.
- A quiz can contain one or more categories of questions. Each category can be assigned to users who answer only the questions in that category.
- The system can send email notifications to these users:
  - Recipients: The recipient can receive notification of an assigned quiz, a quiz whose allowed duration is at 50%, and a quiz that is overdue.
  - Recipient's manager: The recipient's manager can receive notification when a quiz is overdue.
  - Quiz manager: The quiz manager can receive notification of an overdue quiz to which he or she is assigned.

- Quizzes can contain questions that are scored or not scored. Unscored questions assess opinions or involve dates and are not counted in the final score. Scored questions specify correct answers and are scored either as 0% or 100%. You can apply a weighting scale to scored questions to establish their relative importance. You can designate questions with these data types as scored questions:
  - Checkbox
  - Choice
  - Duration
  - Likert Scale
  - Numeric Scale
  - Template
  - Yes/No

- A quiz question can be dependent on the response to any scored question. For example, you can create a dependent question requesting additional information that appears only if a recipient answers No to a specific question.

Quiz application: Important terms

The quiz application involves several terms.

<table>
<thead>
<tr>
<th>Table 147: Terms used in quiz application</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quizzes</strong></td>
</tr>
</tbody>
</table>
Categories

A quiz category represents a theme for quiz questions. Each category contains one or more questions and names the recipients for the questions in that category. By default, the system creates one category with the same name as the quiz. You can create additional categories as needed. Categories can be weighted higher or lower to determine the importance of that category in the overall score.

Questions

A quiz question is a question configured for a category and sent only to the users for that category. Questions have a wide variety of data types and can be individually weighted higher or lower. Questions may be scored or unscored.

Category user

A category user is the recipient of questions for a specific category. You can select different users to answer the questions for each category.

Templates

A template is a question data type that provides reusable rating scales for answers to questions. For example, the answer template named Satisfaction contains a satisfaction scale ranging from Very Satisfied to Very Dissatisfied.

Set up and administer quizzes

Set up and administer quizzes.

Role required: assessment_admin or admin

1. Optional: Create reusable question templates for quizzes.
2. Create the quiz using the Quiz Designer or with forms accessed from the navigation menu.
   The quiz record includes specifics such as duration, notification preferences, a questionnaire introduction, and ending notes displayed to recipients.
3. Edit the default category or create additional categories as needed.
   The system creates a category with the same name as the quiz.
4. Define users for each category.
   These are the recipients who answer the questions in a category. You can define different users for each category.
5. Create the questions for each category.
6. Create the answers for each question and determine if the questions are scored.
   You can create unique answers or select preconfigured answers from a template.

Publish and distribute the quiz. You can send the quiz to a single user or all users in each category.
Review the results from the submitted quizzes in reports and scorecards.

Quiz roles

The Quizzes application uses these roles. No role is required to take quizzes that are assigned to you.
### Table 148: Quiz roles

<table>
<thead>
<tr>
<th>Role Title [Name]</th>
<th>Role Description</th>
</tr>
</thead>
</table>
| assessment administrator [assessment_admin]   | Can administer the Assessments application and all quiz records. Can access all the modules of the Assessments application.  

**Note:** The itil_admin role and the survey_admin role contain the assessment_admin role. |
| Administrator [admin]                          | Can access all aspects of the assessment and survey processes. Only administrators can modify survey notifications, create survey modules, and import surveys.                                                     |

### Activate the quiz designer

Administrators can activate the Quiz Designer plugin.

Role required: assessment_admin or admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.  
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.  
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

### Quiz Overview module

The Quiz Overview module is a homepage that displays various gauges that report on data such as results for each category and quizzes that are complete, pending, or in progress.

Role required: assessment_admin or admin

For details about editing gauges on homepages, see Add existing gauges to a homepage on page 478.

Users with the assessment_admin role can view the overview page and refresh, add, delete, and rearrange gauges.

1. Navigate to Quizzes > Overview.
2. Click elements within the gauges to obtain more information. The available gauges are:

<table>
<thead>
<tr>
<th>Gauge</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes by State</td>
<td>Assessment Instance [asmt_assessment_instance]</td>
</tr>
<tr>
<td>Total Questions by Quiz</td>
<td>Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td>Questions by Data Type</td>
<td>Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td>Correct Answers by Assigned User</td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
</tbody>
</table>
Quiz designer

The quiz designer provides a single interface that users with the assessment_admin role can use to create, edit, and distribute quizzes.

You can also use it to edit existing quizzes and change scoring parameters.

Alternatively, you can use the modules of the assessment engine to create and edit the records that make up a quiz. All quiz records are stored in assessment tables and displayed in Quiz views of those tables.

Tools on the Quiz Designer

The quiz designer includes a design canvas, a header bar, and many controls that you can use to create quizzes.

To open the quiz designer, navigate to Quizzes > Quiz Designer.

The designer contains the following elements:

• Controls
• Header bar
• Design canvas

Controls

Controls for the supported question data types are available in the Controls palette. Drag and drop a control onto the designer canvas to create a question of that type.

<table>
<thead>
<tr>
<th>Gauge</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Correct Answers</td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Incorrect Answers by Assigned User</td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Number of Incorrect Answers</td>
<td>Metric Results [asmt_metric_result]</td>
</tr>
</tbody>
</table>
Table 149: Question controls

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
<th>Scored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Question with a check box or a Yes/No list for user responses.</td>
<td>Y</td>
</tr>
<tr>
<td>Choice</td>
<td>List of predefined options. For more information, see the definition for Choices Create quiz questions on page 403.</td>
<td>Y</td>
</tr>
<tr>
<td>Date</td>
<td>Date field.</td>
<td>N</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Date and time field.</td>
<td>N</td>
</tr>
<tr>
<td>Number</td>
<td>Number field with predefined minimum and maximum values. The default is 1-10.</td>
<td>N</td>
</tr>
<tr>
<td>Percentage</td>
<td>Percentage field with a prescribed range.</td>
<td>N</td>
</tr>
<tr>
<td>Data type</td>
<td>Description</td>
<td>Scored</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Scale</td>
<td>Predefined Likert scale. Answer options appear as radio buttons.</td>
<td>Y</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>Selectable number scale. The default is 1-5. Answer options appear as radio buttons.</td>
<td>Y</td>
</tr>
<tr>
<td>String</td>
<td>Single or multiline text field.</td>
<td>N</td>
</tr>
<tr>
<td>Template</td>
<td>Choice list of templates that provide a predefined scale of options. For details, see Configure a template question.</td>
<td>Y</td>
</tr>
</tbody>
</table>

**Header Bar**

The header bar contains tabs that display different views and a menu of various functions.

Click one of the following tabs to change the view in the canvas:

- **Design**: Add categories and questions, and configure the properties of each. This is the default view of the canvas when you open the designer.
- **Configuration**: Create introductions and end notes for quizzes, and select a signature.
- **Availability**: Select the recipients for each category in the quiz.

Point to the menu icon (menu) in the in the upper right of the quiz designer to select the following options:

- **Save**: Saves the current quiz.
- **Preview**: Displays a preview of the quiz as it appears to the recipients.
- **Publish**: Distributes the quiz to the selected recipients.
- **Save and Publish**: Saves and distributes the quiz in one step.
- **New Quiz**: Opens a fresh canvas for a new quiz.
- **Load Quiz**: Opens a list of existing quizzes that you can select and edit.

The availability of each option depends on the status of the quiz that is opened in the quiz designer.

**Design Canvas**

New quizzes open in the **Design** view. The quiz **Name** field appears above first category in the canvas. A blank question field appears in the category container.

*Create a quiz*

When you create a quiz, you can create one or more categories and then add questions to each category.

Role required: assessment_admin or admin

Each category can be assigned to a different user or the same users. You can also customize each question and make it dependent on the response to another question.

Create a quiz using these procedures:

Create quiz categories

A category represents a theme for evaluating a specific element of the quiz topic and contains questions pertaining to that theme.

Role required: assessment_admin or admin
When you create a quiz, the system creates a default category, using the name of the quiz. You can use this category, modify it, or create additional categories as needed. To have any results, a category must contain scored questions.

1. Navigate either Quizzes > Quiz Designer or Quizzes > Quizzes and click Quiz Designer in the list header.

2. Enter the name of the quiz in the Name field. The system uses this name as the name of the quiz and the first category.

3. To configure the category, click the gear icon in its title bar.

   The Properties dialog box appears. You can change the name of the category, add a description for it, and enter text in the Details field that introduces or explains the category to the recipients. The system updates the category as you type.

4. Click the X icon to close the category properties dialog box and save your settings.

5. To add a new category, click the + icon in the title bar of an existing category.

Create quiz questions

You can create multiple questions for each category but each question can be associated with only one category.

Role required: assessment_admin or admin

The data type that you select for each question determines how it can be answered by quiz recipients.

You can designate questions to be scored. Only scored questions are shown in the quiz results and considered when calculating the category results. You must also specify a correct answer for scored questions.
Note:
To designate a question as scored, you must use Assessment forms. For instructions, see *Configure a scored question*.

1. In the Design view, drag a data type icon from the Controls palette and drop it into a category container.

2. To configure the question, click in the gear icon in its title bar. The Properties dialog box appears for the question.
3. Fill in the fields on the form, as appropriate.

Table 150: Question property fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>[Required] Name of the question. Create a concise and easily recognizable name for your question. The system uses this value to identify the question in Assessment Metric lists and in scorecard charts.</td>
</tr>
<tr>
<td>Question</td>
<td>Text to display as the question on quizzes. Enter a clear, straightforward question that is easy to understand.</td>
</tr>
<tr>
<td>Type</td>
<td>[Read-only] Data type selected for this question. See the table in Controls for possible data types.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that determines whether this question is available on a quiz. If a question is marked inactive, it does not appear on quizzes generated after the question becomes inactive.</td>
</tr>
<tr>
<td>Boolean option</td>
<td>Whether a check box or a Yes/No list appears as the option for the Boolean question.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Check box for requiring users to answer the question. Users cannot submit quizzes until they answer all mandatory questions, which are denoted by a red field status indicator. This field is available when the question does not have a dependency and the question Controls is not Boolean with a check box option.</td>
</tr>
<tr>
<td>String option</td>
<td>Setting for the appearance of a string field in a question. This field is available when the question type is String. The string options are:</td>
</tr>
<tr>
<td></td>
<td>• Single line: Single line text field 40 characters in length that allows strings of any length.</td>
</tr>
<tr>
<td></td>
<td>• String line wide: Full page width text field that allows a single line entry of any length.</td>
</tr>
<tr>
<td></td>
<td>• Multiline: Full page width multi-line text field that allows word wrap and returns.</td>
</tr>
<tr>
<td>Min</td>
<td>Lowest positive whole number that users can enter or select to answer the question. This field is available when the question type is Number, Percentage, or Numeric Scale.</td>
</tr>
<tr>
<td>Max</td>
<td>Highest positive whole number that users can enter or select to answer the question. This field is available when the question type is Number, Percentage, or Numeric Scale.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allow not applicable</td>
<td>Check box for including <strong>Not Applicable</strong> as an option for this question. Users can select <strong>Not Applicable</strong> if they do not have sufficient information to respond to a question. User responses of <strong>Not Applicable</strong> are excluded from results calculations. This field is available when the question does not have a dependency and the question <strong>Date type</strong> is not <strong>Boolean</strong> with a check box option.</td>
</tr>
<tr>
<td>Randomize answers</td>
<td>Check box for displaying answer options in a random order whenever the question appears in a quiz. Answer preference is sometimes influenced by the order in which options appear, which can result in biased results. Randomizing options can help prevent this bias. <strong>Note:</strong> Randomizing options for certain questions may make those questions confusing for users. In general, only randomize options that do not follow a logical order.</td>
</tr>
<tr>
<td>Details</td>
<td>Information about the question that is displayed on the quiz. Include details that help users understand how to answer the question. You can also enter HTML text in this field with the WYSIWYG editor. For example, include HTML to embed links and images.</td>
</tr>
<tr>
<td>Correct answer</td>
<td>Answer option that you want to be selected by users. When you specify a correct answer for a question, the system scores the question. This field is available for all data types except <strong>Date, Date/Time,</strong> and <strong>String.</strong></td>
</tr>
</tbody>
</table>
### Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choices</td>
<td>Options for a question with a data type of <strong>Choice</strong> or <strong>Scale</strong>. The system automatically adds text and values that you can edit for each option. Click the <strong>+</strong> icon to add an option, or click the <strong>X</strong> icon to delete an option. By default, the system arranges options in the order established by their values. To change the order, drag and drop the options.</td>
</tr>
</tbody>
</table>

### Dependency

| Displayed when | Condition builder that hides or displays the question depending on the answer to another question in the same category. Select an existing question from the list with a data type of **Boolean**, **Choice**, **Scale**, or **Template**. Create the condition that must exist for recipients to see the dependent question, using the **is** or **is one of** operator. The system prevents recursive dependencies between questions. For example, if Question A depends on Question B, Question B cannot depend on Question A. |

4. To create any special conditions that must be met for a question to appear on the quiz, click the **Dependency** tab. Fill in the fields, as appropriate.

   Dependent questions appear on the quiz when a recipient selects a specific answer or combination of answers to another question in the same category

5. Select a question in the Displayed when field. The system selects the appropriate operator and displays the possible answers for the selected question.

6. Select the answers that satisfy the condition. Selected answers are indicated by a check mark.
7. Click X to close the question properties dialog box and save your settings.

8. To add a question with the same data type as an existing question in the category, click the + in the title bar of the existing question.

9. Drag questions to change their order within a category or move them between categories.

10. To delete a question, click the X in its title bar.

Configure a template question
You can configure template questions when designing quizzes.

Role required: assessment_admin or admin

1. Drag the Template data type icon into a category container.
2. Click the gear icon in the question title bar to open the template properties dialog box.
3. Select a predefined scale from the list.
Question entry fields appear for that template.

4. Enter one or more questions that are appropriate for the template.
5. Click the arrow to the right of a question to configure its properties. You must provide a name for each question.
6. Click the up or down arrow to move between questions, or click the back arrow to return to the template properties dialog box.

7. Configure the properties for the remaining questions.

8. Click the X icon to close the template properties dialog box and save your settings.

Configure a scored question

Only scored questions are considered when calculating category and quiz results.

Role required: assessment_admin or admin

A question must have a correct answer specified to be scored. Only results for scored questions are displayed in the quiz scorecard.

1. Navigate to Quizzes > Quizzes.

2. Open the quiz containing the questions you want to mark as scored.

3. In the Metric Categories related list on the Assessment Metric Type form, select the category for the questions you want to mark as scored.

4. In the Assessment Metrics related list on the Metric Category form, select a question from the list.

5. In the Question Type section of the Assessment Metric form, select the Scored check box.
This check box is not available if the question's data type is not supported for scoring.

6. Select the **Correct answer** for the question.
7. Click **Update**.
8. Repeat the process for all the questions in the category that you want the system to score.
9. Return to the list of metric categories, select another category, and configure scoring for the appropriate questions in that category.

Configure a quiz

You can configure an entire quiz.

**Role required:** assessment_admin or admin

The values you enter and select on this page are applied to the entire quiz.

In the quiz designer, click **Configuration** and then fill in the fields as described in the table.

---

**Table 151: Quiz designer configuration**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Check box for enabling the distribution of this quiz to recipients.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of this configuration or the quiz to which it is attached.</td>
</tr>
<tr>
<td>Introduction</td>
<td>Introductory content to display on quizzes. You can add a welcome message or background information about the quiz.</td>
</tr>
<tr>
<td>Signature</td>
<td>[Optional] Acknowledgement by a quiz recipient of requirements, admonitions, or expectations related to a quiz.</td>
</tr>
<tr>
<td>Return URL</td>
<td>Destination address of a web page that is presented to users after they submit a completed quiz. When a return URL is configured, the <strong>End note</strong> content does not appear.</td>
</tr>
<tr>
<td>End note</td>
<td>Content that is displayed to recipients after they submit a completed quiz. You can add a thank you message, follow-up instructions, or other applicable information. End notes are not displayed if a <strong>Return URL</strong> is specified.</td>
</tr>
<tr>
<td>Duration</td>
<td>Amount of time that recipients are given to complete this quiz, starting from the time that the quiz is generated. The default duration is 14 days.</td>
</tr>
<tr>
<td>Manager</td>
<td>Assessment manager for this quiz. These users are only responsible for managing the quiz process and not the results. The system notifies the manager when submissions for this quiz are past due.</td>
</tr>
</tbody>
</table>
Select a quiz recipient
When the system distributes a quiz, it sends email notifications to the category users and their managers.

Role required: assessment_admin or admin

A category can have one or more assigned users, and the same user can be assigned to more than one category.

The system also creates a link to the quiz in the recipients’ My Assessments & Surveys portal. Users can only answer questions in the categories that they are assigned to.

To select the recipients for each quiz category, click Availability and then fill in the fields as described in the table.

Table 152: Select recipients

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select category</td>
<td>Category for which the selected users are recipients.</td>
</tr>
<tr>
<td>Add users</td>
<td>Users selected as recipients for this category. The choice list shows all users in the system.</td>
</tr>
</tbody>
</table>

Publish a quiz
When you publish a quiz, the system sends email notifications to the recipients and to their managers. A card in each recipient's My Assessments & Surveys portal displays a link to the quiz.

Role required: assessment_admin or admin

1. If you want to see the quiz as the recipients will see it, point to the menu icon in the upper right of the quiz designer and click Preview.
2. When you are satisfied with the quiz, click Save and Publish or Publish to distribute it.

Create quizzes with forms
As an alternative to the Quiz Designer, you can create a complete quiz using records in the Assessment application.

All the elements of a quiz, the categories, questions, and answers, are stored in tables used by the assessment engine and are displayed in quiz views of these tables. Users creating quizzes in the Assessment application must have the assessment_admin role.

The best practice for creating a quiz using assessment forms is to follow the procedures in the order shown here:

- Create the quiz.
- Set up the categories.
- Create the questions for the quiz.
- Create the answers for the questions.
• Distribute the quizzes to recipients.

**Note:** The recommended method of creating and editing quizzes is to use the quiz designer, which provides a single, intuitive interface for creating and editing quizzes quickly. If you determine that you need to add specific features to your quiz not offered through the quiz designer, you can do so by using some of the specific procedures described here.

---

**Edit a quiz**
You can update a quiz after the quiz has been distributed.

Role required: assessment_admin or admin

• Questions that you add are available only on quizzes that are distributed after the update.

• Before a quiz is submitted or during the retake period:
  • Changes to existing questions are immediately available to users. This includes changes to the answers, such as additional choices or changes to the data type.
  • Deleted questions are deleted from distributed quizzes in users' queues.

Open the quiz: Group the Service Details form with one of the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes &gt; Quiz Designer</td>
<td>Point to the menu icon in the quiz header bar and select Load Quiz.</td>
</tr>
<tr>
<td>Quizzes &gt; Quizzes</td>
<td>Right-click a quiz in the list and select Quiz Designer.</td>
</tr>
<tr>
<td>Quizzes &gt; Quizzes</td>
<td>Open a quiz from the list and click Quiz Designer in the header of the Assessment Metric Type form.</td>
</tr>
</tbody>
</table>

**Enable a quiz retake**
You can configure a quiz to allow recipients to resubmit their answers as many times as they like, until the quiz’s due date.

Role required: assessment_admin or admin

Results are not calculated until the quiz’s configured duration has elapsed. The card in the user’s queue remains visible until the quiz’s due date and displays a button to allow retakes.

1. Navigate to Assessments > Metric Definition > Types.
2. Remove the Evaluation method = Assessment filter condition so you can see all the records in the list.
3. Open the quiz.
4. In the Assessment Metric Type form, select the Allow retake check box and save the record.

**View a quiz result**
You can view quiz results for each question and category, or view the quiz scorecard for a detailed breakdown.

Role required: assessment_admin or admin

Quiz results are stored in the Metric Result [asmt_metric_result] table and display recipients' answers to each question in a category.

Navigate to Quizzes > Quiz Results.
View a quiz designer scorecard

The quiz scorecard opens in the Category Results view.

For detailed information about scorecards, see Quiz Scorecards.

1. Navigate to Quizzes > Quizzes.
2. Open the quiz whose results you want to view.
   
   The scorecard for the selected quiz opens in the Category Results view.
4. Select a category to view from the choice list above the chart.
View a quiz category result

Category results are stored in the Assessment Category Result [asmt_category_result] table and display the overall ratings for each category based on the weighted value for each scored question.

1. To view category results, navigate to Quizzes > Category Results.

2. The system calculates results from the weight configured for each category. Weights are set to a value of 10 by default but can be changed. To edit a category's weight:
   3. Navigate to Quizzes > Categories, and select a category from the list.
   4. In the Metric Category form, edit the default value in the Weight field.
   5. Click Update.

   **Note:** You can also edit the weight of a question in the Quiz view of the Assessment Metric form.

Take a quiz

All available quizzes assigned to you appear in your personal assessment and survey queue. When you submit a quiz, the system stores your responses. If a quiz allows retakes, you can resubmit it as many times as you like until its due date.

Role required: none

No special role is required to complete a quiz, but you must be logged in and the quiz must be assigned to you.

If configured to send email, the system sends you a notification when a quiz is assigned to you. The message contains a link to the quiz and instructions for viewing the assessment and survey queue.

Each quiz assigned to you appears in your assessment and survey queue as a card that contains information about the quiz, including a launch button. Overdue quizzes are marked with a red icon and red due date.

You must answer every required question, indicated by a red star, before you can submit the quiz. If you start to take a quiz but cannot complete it, save your responses and return to it later. When you have answered all the questions and are satisfied with the responses, submit the quiz.

By default, you cannot modify your answers to a quiz after submission. However, if the administrator has configured this quiz to allow retakes, the quiz remains in your queue after you finish it and the card shows a Modify Quiz button. You can retake the quiz as many times as you like before the due date.

1. Navigate to Self-Service > My Assessments & Surveys.
2. Click **Take Quiz** on a quiz card to open the questionnaire. If there is more than one quiz category, you can click the collapse or expand icon to hide or show the questions in the category.

3. Answer each question to the best of your ability. If you are unsure of how to respond to a question or if a question does not apply to you, select Not Applicable, if that choice is available.

4. Click **Save** at any time to save your responses without submitting them. You can reopen the quiz from your queue when you are ready to work on it again.

5. When you are ready to submit the completed quiz, read any assertions that require your attention.

6. If present, select the check box to acknowledge the assertion. If your full name is displayed, you are required to provide authentication to acknowledge the assertion after you submit the quiz.

7. Click **Submit**.

8. If prompted, enter your user name and password to verify your signature. If you provided valid answers for all mandatory questions, a success message appears, displaying any end note that was configured. If the quiz allows retakes, the card remains in your queue with a **Modify Quiz** button after submission. If retakes are not permitted, the card disappears from your queue.

9. If the system displays an error message indicating that a question has an invalid response or must still be answered, correct the error and resubmit the quiz.

10. To edit your answers and resubmit a quiz that permits retakes, click **Modify Quiz**. You can modify your responses to the quiz until its due date.

**Quiz scorecards**

The Quizzes application prepares printable scorecards.

A scorecard analyzes category and question responses and compares current ratings with previous ratings. You can examine ratings over time, compare question ratings, or compare the ratings of all categories. All ratings are averages for the time range selected.

The system dynamically updates a scorecard each time you view it, so the ratings reflect recently completed quizzes.

**Category results**

The Category Results view is a stacked bar chart of responses to all questions in a category.

Select the category to display from the choice list above the chart. Category results are only calculated for scored questions

This view displays responses that use the following **data types**:

- Checkbox
- Choice
- Likert Scale
- Number
- Template
- Yes/No

**Note:** The **Checkbox** and **Yes/No** data types are combined into the **Boolean** data type in the **Quiz designer**.
Figure 203: Quiz category results
To view details about a specific response to a question, point to the colored segment representing a specific response. The chart displays the count for those responses and the percentage it represents of the total responses to that question.

**Figure 204: Quiz category result details**

*Question results*
The Question Results view shows the results for all questions in a quiz.
Select a question by name from the choice list to display the results in a pie chart or a bar chart, based on the data type.

**Pie chart**
The pie chart shows question results for these *data types*:
- Checkbox
- Choice
- Likert Scale
- Number
- Template
- Yes/No

*Note: The Checkbox and Yes/No data types are combined into the Boolean data type in the Quiz designer*
Figure 205: Quiz scorecard question results - pie chart
**Bar chart**

A bar chart appears when question results use this data type:

- Percentage

By default, all results for percentage questions use a report range of 20% segments. To configure a report range, navigate to **Reports > Administration > Report Ranges**.

**Average ratings**

The Average Ratings view displays a bar chart of the weighted average rating for each question in a category.

Use this view to learn how individual questions affect the overall rating for the category. Select a category from the second choice list above the chart. Ratings are only calculated for *scored questions*.
Figure 206: Average Ratings view

To view the effect of each question's ratings on the entire category's ratings, point to the colored bar. The pop-up box shows the percentage of the total ratings represented by each individual question's weighted average.
Quiz reports

Quizzes provide several global reports so that assessment administrators can view important statistics. You can share these reports with specific users or groups and change the display options. For detailed field information and reporting options, click the link for the chart Type. To sort a column in ascending or descending order, click the arrow in the column heading.

Table 153: Quiz reports

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes by State</td>
<td>State of distributed quizzes. This chart displays the percentage of distributed quizzes that are ready to take, in progress, complete, or canceled.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Pie Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Assessment Instance [asmt_assessment_instance]</td>
</tr>
<tr>
<td>Total Questions by Quiz</td>
<td>Total number of questions for all categories in each quiz.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td>Questions by Data Type</td>
<td>Total number of questions in all quizzes by data type.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Assessment Metric [asmt_metric]</td>
</tr>
<tr>
<td>Correct Answers by Assigned User</td>
<td>Total number of scored questions answered correctly by each assigned user.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Number of Correct Answers</td>
<td>Total number of correct answers for each scored question.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Incorrect Answers by Assigned User</td>
<td>Total number of scored questions answered incorrectly by each assigned user.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
<tr>
<td>Number of Incorrect Answers</td>
<td>Total number of incorrect answers for each scored question.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Bar Chart</td>
</tr>
<tr>
<td></td>
<td>• <strong>Table</strong>: Metric Results [asmt_metric_result]</td>
</tr>
</tbody>
</table>

**Wizards**

Administrators can create wizards to provide a step-by-step sequence of dialog boxes that lead the user through a procedure. For example, a wizard may lead users through reporting an incident (creating an incident record) without opening a list or form.

![Figure 208: Record generator wizard](image)

Create a basic wizard

This example creates a basic wizard that reports an incident.

1. **Activate the system wizards application**, if necessary.
2. Navigate to **System Wizards > Wizards**.
3. Click **New**.
4. Enter the wizard Name of **Report Incident**.
5. Right-click the header and select **Save**.

Define a wizard variable

How to define a wizard variable.

**Note:** To learn more, see *Define a Wizard Variable*.

Define wizard variables:

![Wizard Variable](image)

Figure 209: Basic variables

1. In the Wizard Variable related list, click **New**.
2. Enter the variable details.
• Type: Multiple Choice
• Name: incident
• Question: What are you having difficulties with?

3. Right-click the header and select Save.
4. In the Question Choices related list, click New.
5. Enter Email Issue in the Text and Value fields, and click Submit.
6. Repeat steps 4 – 5 for the following question choices:
   • Computer Issue
   • Password Reset

7. In the Wizard Variable related list, click New.
8. Enter the variable details and click Submit.
   • Type: Wide Single Line Text
   • Name: description
   • Question: Please describe your symptoms

Create the first panel

Create the first panel of the example wizard.

Note: To learn more, see Wizard Panels.

1. In the Wizard Panels related list, click New.
2. Select a Type of A panel that prompts the user to answer questions.
3. Enter the panel details.
   • Name: Service Questions
   • Title: Service Desk Wizard

4. Right-click the header and select Save.
5. In the Variables related list, click Edit....
6. Using the slushbucket, select and arrange the variables as listed:
   1. What are you having difficulties with?
   2. Please describe your symptoms

7. Click Update.
Create the second panel option

How to create the second panel option.

1. In the Wizard Panels related list, click **New**.
2. Select a Type of **A panel that creates something (like a change request or an incident)**.
3. Enter the panel details and click **Submit**.
   - **Name**: Email Incident
   - **Table**: Incident [incident]
   - **Final View**: ess
4. Right-click the header and select **Save**.
5. In the Field Setters related list, click **New**.
6. Enter the field setter details and click **Submit**.
   - **Type**: Set field to a variable
   - **Field**: Short description
   - **Variable**: description

7. In the Field Setters related list, click **New**.

8. Enter the field setter details and click **Submit**.
   - **Type**: Set field to a specific value
   - **Field**: Category
   - **Value**: software

9. Repeat steps 1 – 8 to create additional panels with the values listed in the table.

<table>
<thead>
<tr>
<th>Step 3. Panel Name</th>
<th>Step 8. Category Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Incident</td>
<td>hardware</td>
</tr>
<tr>
<td>Password Incident</td>
<td>network</td>
</tr>
</tbody>
</table>

**Test the wizard**

Run through several scenarios to test transition logic and record generation.

1. Navigate to **System Wizards > Wizards**.
2. Click **Report Incident**.
3. Click **Test**.
4. Enter the following information and click **Next**.
   - What are you having difficulties with?: Email Issue
   - Please describe your symptoms: Can't access email
5. Verify that a new record is created and the values match step 4.
6. Repeat steps 1 – 5 to test each multiple choice answer.
Wizard transitions

Transitions define logic used to move between panels in wizards.

Transition logic may be based on a predefined order (basic panel flow) or user input (defined transitions).

**Note:** The System Wizards application is not active by default.

*Use basic panel flow in a wizard*

Basic panel flow moves through wizard panels in order, without the need to define logic for each transition. A wizard may use either basic panel flow or defined transitions, but not both.

1. Open the wizard.
2. Select the **Basic panel flow** check box. If necessary, configure the wizard form to add the field.
3. Enter the **First panel** in the reference field.
4. For each panel, define an **Order**. If necessary, configure the panel form to add the field.

*Define a panel transition*

1. Open the panel.
2. In the **Transitions** related list, click **New** or select the transition to edit.
3. Using the **condition builder**, define a transition condition based on wizard variables. Leave the condition empty to use the transition in all cases.
4. In the **To** field, select the next panel.
5. In the From field, select the previous panel.
6. Optional: Define a Transition script that runs when the transition is used.

### Wizard variables

Wizard variables are questions that collect and store user input. Define variables for a wizard and then add them to wizard panels.

Use data collected by wizard variables to:
- Define transition conditions
- Set field values in record generators
- Define dynamic functionality with UI policy and client scripts
- Implement advanced functionality with wizard scripts

**Note:** The System Wizards application is not active by default.

---

**Add a variable to a panel**

To collect user input, add variables to a panel that prompts user to answer questions (wizard panel) or a survey panel (requires Survey Wizard plugin):

1. Navigate to **System Wizards > Wizards**.
2. Open the wizard.
3. In the **Wizard Panels** related list, open the panel to which you are adding a variable.
4. In the **Variables** related list, click **Edit**.
5. Using the **slushbucket**, add and arrange the order in which variables appear on the panel. Only variables defined on the wizard are available on a panel.
Define a question choice
Some variable types require choices. For example, a multiple choice question—such as What kind of e-mail account do you want?—requires options—such as Exchange and UNIX.

1. Open the variable definition.
2. In the Question Choices related list, click New.
3. Enter the question choice details and save the record.
   • Text - option the user sees
   • Value - value stored in the variable
4. Repeat steps 2 – 3 for all available options.

Define a wizard variable
1. Navigate to System Wizards > Wizards.
2. Open the wizard to which you are adding a variable.
3. In the Wizard Variable related list, click New.
4. Select the variable Type.
   Wizards use the same variable types as service catalog items.
5. Enter the name used by the system, for example: `resolved_to_satisfaction`.
6. Enter an order number and select the name of the associated Wizard in the **Expert** field.
7. Enter a descriptive question in the **Question** field.
8. Enter the remaining variable details if necessary and save the record.

### Publish the basic wizard

Create a wizard launcher to make the wizard available through the service catalog homepage.

**Note:** To learn more, see *Publish a Wizard.*
Figure 210: Basic publish

1. Navigate to **Service Catalog > Wizards**.
   If you do not see **Wizards** under **Service Catalog**, complete the next step.

2. Optional: To add the **Wizards** module to the **Service Catalog**, perform the appropriate action for your version of the UI:

   | UI16         | 1. Navigate to **System Definition > Application Menus**.  
   |             | 2. Select **Service Catalog**.  
   |             | UI15 or UI11  
   |             | Right-click the **Service Catalog** application menu and select **Edit Application Menu**.

3. Click **New**.

4. Enter the wizard launcher details and save the record.
   - Name - Report an Incident
   - Category: Can We Help You?
   - Wizard: Report Incident
   - Active: select the check box

*Link to a Wizard*
Provide access to wizards via links, such as defining a new module or sending a link via email (public wizards only).

<table>
<thead>
<tr>
<th>Desired Action</th>
<th>URL schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start a new wizard or resume an in progress wizard if the user is logged in</td>
<td>https://&lt;base URL&gt;/nav_to.do?uri=expert_shell.do?sysparm_sys_id=&lt;wizard sys_id&gt;</td>
</tr>
<tr>
<td>Require the user to restart a wizard from the beginning regardless of previous progress</td>
<td>https://&lt;base URL&gt;/nav_to.do?uri=expert_shell.do?sysparm_sys_id=&lt;wizard sys_id&gt;%26sysparm_initial=true</td>
</tr>
</tbody>
</table>

Example - Define a Wizard Module in Self-Service:

1. Navigate to **System Wizards > Wizards**.
2. Open the wizard.
3. In the **Roles** field, add the user roles that have access to the wizard.

4. Right-click the header and copy the **sys_id** for the wizard.

5. Click **Update**.

6. In the navigation pane, right-click **Self-Service** and select **Edit Application**.

7. In the **Modules** related list, click **New**.

8. Enter the following information and save the record.
   - **Title** and **Order**
   - **Link type**: URL (from arguments)
   - **Arguments**: /expert_shell.do?sysparm_sys_id=<wizard sys_id from step 2>

---

**Create a public page**

A public page is accessible to all users without having to log in or have a particular user role.

Create a **public page**:

1. Navigate to **System Definition > Public Pages**. (You may need to enable the module, as it is not active by default.)

2. Click **New**.

3. Enter the name of the UI page specified (for example, **test_page**)

---

**Make a wizard public**

Public wizards are available for use without logging in.

Reasons to make a wizard public include:

- Allowing non-users to report an incident
- Collecting anonymous survey responses (requires Survey Wizard Plugin)

Once a wizard is made public, deliver it using a link via email or a module on the welcome page (available before a user has logged in).

---

**Make a role public**

Add public to all relevant roles for the wizard:

1. Add **public** to the **Roles** field.

2. For each **wizard variable** defined for that wizard, add **public** to the **read roles** and **write roles** fields. (You may need to personalize the form to add the **read roles** and **write roles**.)

3. Similarly, add **public** to the **read roles** and **write roles** fields for each variable associated with each **wizard panel** for that wizard.

---

**Modify a script**

*Modify the script* for all **Record Generators** (wizard panels) within the wizard.

1. Navigate to **System Wizards > Wizards**.

2. Open the relevant wizard.

3. In the **Wizard Panels** related list, open the record generator panel(s).

4. Enter the following script in the **Script** field for each record generator (Personalize the form to add the **Script** field, if necessary):

   ```javascript
   if ( typeof gs. getRoles () == "undefined" || gs. getRoles () == '' ) {
   current. insert () ;
   ```
Process description
Enable and Select the Wizard

Select the wizard to be made public:

1. **Activate** the Wizards application in the application navigator.
2. Navigate to **System Wizards > Wizards**
3. Select and open the wizard to be made public.

Publish a wizard
Administrators can provide users with access to wizards:

- Using links
- As an item in the service catalog

Administrators can also restrict user access to wizards by role and make wizards available to the public (users that are not logged in).

**Note:** The System Wizards application is not active by default. To use wizards, see **Activate a System Wizard.**

To define the user roles that have access to a wizard:

1. Navigate to **System Wizards > Wizards.**
2. Open the wizard.
3. In the **Roles** field, define the user roles that have access to the wizard. Leave the field blank to allow access for all users.

Send a public wizard link via email
After a wizard is configured as public, it can be delivered via email to anyone, including recipients who are not ServiceNow users.

To send a link to the public wizard via email:

1. Navigate to **System Wizards > Wizards.**
2. Open the wizard.
3. Right-click the header and copy the sys_id for the wizard.
4. Create an email with an email link, using the URL schema:
   
   https://<base URL>/nav_to.do?uri=expert_shell.do?sysparm_sys_id=<wizard sys_id from step 3>

Test the page
Verify the wizard is functional for a public user.

Log out, then click the module (in this example, the Test module under the Self-Service application).

Add a wizard to the service catalog
Create a wizard launcher to make the wizard available through the service catalog homepage:

1. Navigate to Service Catalog > Wizards.
2. Click New.
3. Enter the wizard launcher details and save the record.
   - Name - name by which the wizard appears in the service catalog
   - Category - category under which the wizard appears
   - Wizard - reference to the wizard
   - Active - select the check box
   - Short Description and Description (optional)

Create module
Create Module
Create a public module within the parent application, pointing to the wizard,

1. Copy the sys_id for the wizard: right-click the wizard header, and copy the wizard’s sys_id.
2. In the application navigator, right-click the parent application name (for example, Self-Service) and select Edit Application Menu.
3. Add public to the Roles field.
4. In the **Modules** section, select **New** to create a new module, with the following fields, then select Submit to save the record.

**Table 155: Create Module**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Module title (for example, Test)</td>
</tr>
<tr>
<td>Link type</td>
<td>URL (from Arguments)</td>
</tr>
<tr>
<td>Roles</td>
<td>public</td>
</tr>
<tr>
<td>Arguments</td>
<td>expert_shell.do?sysparm_sys_id=&lt;sys_id of the Wizard&gt;</td>
</tr>
</tbody>
</table>

**Note:** The wizard panel does not link to the beginning by default. To cause the wizard to link to the beginning by default, add &sysparm_initial=true to the URL Argument. For example, /nav_to.do?uri=expert_shell.do?sysparm_sys_id=<wizard sys_id>&sysparm_initial=true.

### Survey wizards

The Survey Wizard plugin creates surveys using wizards.

Advantages versus other survey functionality include the ability to:

- Ask different questions based on responses (dynamic surveys)
- Create multiple page surveys
- Record answers for partially completed surveys

This plugin integrates with the Best Practice - Task Survey Management plugin. Because the plugin relates to the Legacy Surveys application, the plugin is documented on the ServiceNow wiki.

**Configure the survey property**

How to configure the survey properties (apply to all survey wizards and surveys).

1. Navigate to **Survey Wizards > Survey Properties**.
2. Ensure the **Enable the enhanced task survey capabilities. Survey distribution is controlled by Survey Conditions** property is enabled (select the check box).

**Create a page**

How to create the survey pages (wizard panels) and add questions.

1. Navigate to **Survey Wizards > All** and open the survey wizard.
2. In the **Survey Panels** related list, click **New**.
3. Enter the **Name**, **Title**, and **Description**.
4. Right-click the header and select **Save**.
5. In the **Variables** related list, click **Edit**.
6. Using the **slushbucket**, select and arrange survey questions on the panel.
7. Repeat steps 1 – 6 for each page of the survey wizard.

**Note:** To learn more, see **Wizard Panels**.
Create a dynamic effect optional
Create dynamic effects, such as hiding or showing fields on a panel based on answers.

1. Navigate to Survey Wizards > Wizard Policy > Wizard UI Policy or Survey Wizards > Wizard Policy > Wizard Client Scripts
2. Click New.
3. Enter the UI policy or client script details.

Note: To learn more, see Wizard UI Policy and Client Scripts.

Define a transition
Transitions define the logic used to move between pages (wizard panels) in a survey.

Use panels and transition logic to implement multiple page surveys and dynamic effects, such as skipping pages based on answers.

Define transitions:
1. Navigate to Survey Wizards > All and open the survey wizard.
2. Open a survey panel.
3. In the Wizard Panel Transitions related list, click New.
4. Enter the transition details and click Submit.
5. Repeat steps 1 – 4 for all transitions between pages in the survey wizard.

Note: To learn more, see Wizard Transitions.

Create a survey condition
How to create a survey condition that controls when and to whom the survey wizard is sent.

Create a survey condition that controls when and to whom the survey wizard is sent:
1. Navigate to Survey Wizards > Survey Conditions.
2. Click New.
3. Select a Type of Survey Wizard. Only a survey or survey wizard is sent (determined by Type), even if both are defined for the condition.
4. Enter the condition details.
5. Click Submit.

Create the email notification
Create an email notification to deliver the survey wizard.

Role required: admin or survey_admin
1. Navigate to System Policy > Email > Notifications.
2. Filter the list to view notifications with an Event Name of task.send_survey, which is the event that is triggered when survey conditions are met.
3. Open a survey notification in the filtered list or click New.
4. Enter the email notification details. If you are creating a new notification, be sure to adjust conditions and weight to avoid conflicts with existing survey notifications (Event Name of task.send_survey).
5. Click Update or Submit.

Create the survey wizard
How to create the survey wizard.
The following steps provide an overview of building a survey wizard.

1. *Activate the Survey Wizard plugin*, if necessary.
2. Navigate to **Survey Wizards > All**.
3. Click **New**.
4. Enter the survey wizard **Name**.
5. In the **Roles** field, add the **public** role.
6. Right-click the header and select **Save**.

**Define a question**
How to define survey questions (wizard variables).

1. In the **Wizard Variable** related list, click **New**.
2. Enter the variable details.
3. In the **Read roles** field, add the **public** role.
   Configure the form to add the **Read roles** field, if necessary.
4. Click **Submit**.
5. Repeat steps 1 – 4 for all questions in the survey.

**Note:** To learn more, see *Define a Wizard Variable*.

**Installed components**
Fields, tables, a business rule, a script include, an application, a survey wizard panel, and demo data (optional).

The following tables are modified:

<table>
<thead>
<tr>
<th>Display Name (Table Name)</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey Conditions (survey_conditions)</td>
<td>Add fields to determine which survey is sent when conditions are met:</td>
</tr>
<tr>
<td></td>
<td>• Type - either a survey or survey wizard</td>
</tr>
<tr>
<td></td>
<td>• Survey_wizard - reference to applicable survey wizard</td>
</tr>
<tr>
<td>Task Survey (task_survey)</td>
<td>Add fields to track which surveys were sent to users:</td>
</tr>
<tr>
<td></td>
<td>• Type - either a survey or survey wizard</td>
</tr>
<tr>
<td></td>
<td>• Survey_wizard - reference to applicable survey wizard</td>
</tr>
<tr>
<td>Survey Instance (survey_instance)</td>
<td>Add fields to track survey wizard instances along with surveys</td>
</tr>
<tr>
<td>Wizard (expert)</td>
<td>Add field to support new survey wizard table</td>
</tr>
</tbody>
</table>

The following tables are added:
Table 157: New tables

<table>
<thead>
<tr>
<th>Display Name (Table Name)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey Panel (expert_panel_survey)</td>
<td>Adds a survey panel to wizards. Extends the expert_panel table</td>
</tr>
<tr>
<td>Survey Wizard (expert_survey)</td>
<td>Stores survey wizards. Extends expert</td>
</tr>
</tbody>
</table>

The **SurveyUtils** script include is added to record responses during survey panel transitions.

The **Task Survey Events** business rule is modified to include logic for sending survey wizards. Customers who have modified this business rule can not install the updated version automatically.

For information about the link generated from the business rule script, see **Survey Wizard - Multiple Instances** on the ServiceNow Community.

The plugin adds **Survey** as a new type of wizard panel. Survey wizards are created using survey panels.

Survey wizard responses are saved when the user navigates between survey panels (clicks **Next** or **Previous**), as opposed to saving responses only at the end (surveys). This feature allows a logged in user to resume a survey that is in progress and allows survey readers to collect response data for partially completed surveys. To support the ability to resume surveys in progress, survey wizard answers are also temporarily stored in XML in the **expert_instance** table. When the user clicks **Done** on the last panel of the survey wizard, the record in the **expert_instance** table is deleted.

For information about having more than one instance of a given Wizard in play at a time, see **Survey Wizard - Multiple Instances** on the ServiceNow Community.

Demo data is available with this plugin. If you choose to install demo data, a sample survey wizard, called Customer Satisfaction Survey, is added to the database.

**Request survey wizard**

Before activating this plugin, consider the installed components, dependencies, and impact.

- Installed Components - fields, tables, a business rule, a script include, an application, a survey wizard panel, and demo data (optional). For more details, see **Installed Components**.
- Dependencies - Best Practice - Task Survey Management (active by default)
- Impact - adds fields to existing tables, installs the Best Practice - Task Survey Management plugin, and modifies the Task Survey Events business rule. You may choose to install a demo survey with the plugin. The plugin integrates with standard survey functionality, so there is no need to transition existing surveys to survey wizards. However, survey administrators must specify a type (survey or survey wizard) for new survey conditions.

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
</tbody>
</table>
Specify the date and time you would like this plugin to be enabled

| Date and time must be at least 2 business days from the current time. |

**Note:** Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.

| Reason/Comments | Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows. |

3. Click **Submit**.

**Test the survey wizard**
How to test the survey wizard.

1. Navigate to **Survey Wizards > All** and open the survey wizard.
2. Click **Try it**.
3. Answer survey questions and verify the transition logic based on answers, if applicable.
4. At survey completion, click **Done**.
5. Navigate to **Survey Wizards > Survey Instances**.
6. Search for and open the most recent instance of the survey wizard.
7. Verify that your survey answers are properly recorded.
8. Run through several scenarios to test transition logic and data collection.

**Create a survey wizard (demo)**
This example creates a survey using a survey wizard.

This example creates a survey using a survey wizard (requires the **Survey Wizard plugin**). The survey uses dynamic features that are available using wizards.

Create the survey wizard demo
Create the survey wizard.

1. **Activate the Survey Wizard plugin**, if necessary.
2. Navigate to **Survey Wizards > All**.
3. Click **New**.
4. In the Name field, enter Software Needs Analysis.
5. In the Roles field, add the **public** role.
6. Right-click the header and select **Save**.
Define a survey question
How to define survey questions (wizard variables).

1. In the Wizard Variable related list, click New.
2. Enter the variable details as listed in the table.
3. In the Read roles field, add the public role.
   *Configure the form* to add the Read roles field, if necessary.
4. Click Submit.
5. Repeat steps 1 – 4 for all variables in the table.

Table 158: Wizard variables details table

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Question</th>
<th>Additional configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckBox</td>
<td>office</td>
<td>Microsoft Office Suite</td>
<td></td>
</tr>
<tr>
<td>CheckBox</td>
<td>creativesuite</td>
<td>Adobe Creative Suite</td>
<td></td>
</tr>
<tr>
<td>CheckBox</td>
<td>other</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td>needs</td>
<td>Business software you use (select all that apply):</td>
<td></td>
</tr>
<tr>
<td>Single Line Text</td>
<td>otherprod</td>
<td>Please specify:</td>
<td></td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>powerpoint</td>
<td>PowerPoint</td>
<td>Scale min: 1</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>word</td>
<td>Word</td>
<td>Scale min: 1</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>excel</td>
<td>Excel</td>
<td>Scale min: 1</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>publisher</td>
<td>Publisher</td>
<td>Scale min: 1</td>
</tr>
<tr>
<td>Numeric Scale</td>
<td>access</td>
<td>Access</td>
<td>Scale min: 1</td>
</tr>
<tr>
<td>Yes/No</td>
<td>photoshop</td>
<td>Photoshop</td>
<td>Default value: No</td>
</tr>
<tr>
<td>Yes/No</td>
<td>illustrator</td>
<td>Illustrator</td>
<td>Default value: No</td>
</tr>
<tr>
<td>Yes/No</td>
<td>acrobat</td>
<td>Acrobat</td>
<td>Default value: No</td>
</tr>
<tr>
<td>Yes/No</td>
<td>dreamweaver</td>
<td>Dreamweaver</td>
<td>Default value: No</td>
</tr>
</tbody>
</table>
Create a survey page

Create the survey pages (wizard panels) and add questions.

1. Navigate to Survey Wizards > All and select Software Needs Analysis.
2. In the Survey Panels related list, click New.
3. Enter the Name, Title, and Description as listed in the table.
4. Right-click the header and select Save.
5. In the Variables related list, click Edit....
6. Using the slushbucket, select and arrange the variables as listed in the table.
7. Repeat steps 1 – 6 for all panels in the table.

Table 159: Panel variables table

<table>
<thead>
<tr>
<th>Name and Title</th>
<th>Description</th>
<th>Add variables</th>
</tr>
</thead>
</table>
| Describe Needs          | Please complete this survey to help us evaluate our ongoing software needs. | Business software you use (select all that apply):
|                         |                                                                           | Microsoft Office Suite
|                         |                                                                           | Adobe Creative Suite
|                         |                                                                           | Other
|                         |                                                                           | Please specify:
| Microsoft Office        | Please rate the importance of each program to your job, using a scale from 1 to 5 (1 = not important, 5 = very important): | Word
|                         |                                                                           | PowerPoint
|                         |                                                                           | Excel
|                         |                                                                           | Publisher
|                         |                                                                           | Access
| Adobe Creative Suite    | Have you used these programs?                                              | Acrobat
|                         |                                                                           | Illustrator
|                         |                                                                           | Photoshop
|                         |                                                                           | Dreamweaver
|                         |                                                                           | Describe projects for which you use these tools:
| End Note                | Thanks for completing our survey. To order new or upgraded software, visit the service catalog. |
Create a dynamic effect

How to create a dynamic effect.

On the Describe Needs panel, if the user selects **Other**, then the Please specify field should be visible and mandatory.

![Image of Survey wizard UI](image)

**Figure 211: Survey wizard UI**

1. Navigate to **Survey Wizards > Wizard Policy > Wizard UI Policy**.
2. Click **New**.
3. Enter the UI policy details.

<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wizard</td>
<td>Software Needs Analysis</td>
</tr>
<tr>
<td>Reverse if false</td>
<td>select the check box</td>
</tr>
<tr>
<td>Order</td>
<td>select the check box</td>
</tr>
<tr>
<td>Active</td>
<td>select the check box</td>
</tr>
<tr>
<td>Short description</td>
<td>If Other is true, show Specify</td>
</tr>
<tr>
<td>Wizard Conditions</td>
<td>other is true</td>
</tr>
</tbody>
</table>

4. Right-click the header and select **Save**.
5. In the Wizard UI Policy Actions related list, click **New**.
6. Enter the action details and click **Update**.
   - Variable Name: otherprod
   - Mandatory: True
   - Visible: True

Define a transition demo
How to define transitions.

1. Navigate to **Survey Wizards > All** and select **Software Needs Analysis**.
2. Open a survey panel as listed in the table.
3. In the Wizard Panel Transitions related list, click **New**.
4. Enter the transition details and click **Submit**.
5. Repeat steps 1 – 4 for all panel transitions in the table.

<table>
<thead>
<tr>
<th>Panel</th>
<th>Condition</th>
<th>Order</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describe Needs</td>
<td>office is true</td>
<td>100</td>
<td>Microsoft Office</td>
</tr>
<tr>
<td>Describe Needs</td>
<td>creativesuite is true</td>
<td>200</td>
<td>Adobe Creative Suite</td>
</tr>
<tr>
<td>Describe Needs</td>
<td></td>
<td>300</td>
<td>End Note</td>
</tr>
<tr>
<td>Microsoft Office</td>
<td>creativesuite is true</td>
<td>100</td>
<td>Adobe Creative Suite</td>
</tr>
<tr>
<td>Microsoft Office</td>
<td></td>
<td>200</td>
<td>End Note</td>
</tr>
<tr>
<td>Adobe Creative Suite</td>
<td></td>
<td>100</td>
<td>End Note</td>
</tr>
</tbody>
</table>

Test the survey wizard demo

How to test the survey wizard.

1. Navigate to **Survey Wizards > All** and select **Software Needs Analysis**.
2. Click **Try It**.
3. In the Business software you use (select all that apply) question, select: **Microsoft Office, Other**.
4. Verify that the Please specify field appears and is mandatory when Other is selected (UI policy created in **Dynamic Effects**).
5. Click **Next**.
6. Verify that the Office panel appears.
7. Enter test values for the questions and click **Next**.
8. Verify that the End Note panel appears (the transition logic skips the Adobe Creative Suite panel when the Adobe Creative Suite check box is not selected on the first panel).
9. Click **Done**.
10. Navigate to **Survey Wizards > Survey Instances**.
11. Search for and open the most recent instance of the Software Needs Analysis survey wizard.
12. Verify that your survey answers are properly recorded.
13. Run through several scenarios to test transition logic and data collection.
Configure the survey

How to configure the survey.

1. Navigate to Survey Wizards > Survey Properties.
2. Ensure the Enable the enhanced task survey capabilities. Survey distribution is controlled by Survey Conditions. property is enabled (select the check box).
4. Click New.
5. Enter the condition details and click Submit.

Table 162: Condition details

<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Survey Wizard</td>
</tr>
<tr>
<td>Survey wizard</td>
<td>Software Needs Analysis</td>
</tr>
<tr>
<td>Active</td>
<td>select the check box</td>
</tr>
<tr>
<td>Table</td>
<td>Incident</td>
</tr>
<tr>
<td>User field</td>
<td>Caller</td>
</tr>
<tr>
<td>Description</td>
<td>Send software survey to all users with software questions</td>
</tr>
<tr>
<td>Condition</td>
<td>Category is Software</td>
</tr>
</tbody>
</table>

6. Navigate to System Policy > Email > Notifications.
7. Click **New**.
8. Enter the email notification details and click **Submit**.

<table>
<thead>
<tr>
<th>Table 163: Email notification details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Event Name</td>
</tr>
<tr>
<td>Table</td>
</tr>
<tr>
<td>User field</td>
</tr>
<tr>
<td>Active</td>
</tr>
<tr>
<td>Conditions</td>
</tr>
<tr>
<td>Subject</td>
</tr>
<tr>
<td>Message</td>
</tr>
</tbody>
</table>

9. Filter the Email Notifications list using the **condition**: Event Name is task.send_survey and Table is Incident.
10. If any other notifications exist, increase their Weight value so that only the Software Survey is sent when the incident category is software (the Software Survey has a Weight of 0, so it has the highest priority of any notifications for the incident table).

**Wizard panels**

Wizard panels are screens that appear in a defined order. Each step in a wizard is represented by a panel. A wizard panel prompts users to answer questions. Wizard panel type-specific information is:

- **Title** - label that appears above the panel in wizard view.
- **Description** - text that appears above questions in wizard view.
- **Variables** - questions that collect user input. To learn more, see *Wizard Variables*.

**Note:** The System Wizards application is not active by default.

*Catalog order and checkout panels*

What catalog order and checkout panels display.

A catalog order panel displays listings for service catalog items in tabbed view. A catalog checkout panel displays an order confirmation screen for the items on the preceding catalog order panel. Use these panels to create an order guide using wizards.

*Change a banner setting*

How to change a banner setting.

Steps displayed in the banner can be manually defined or automatically generated.

To change the banner settings:

1. Navigate to **System Wizards > Wizards**.
2. Create or open a wizard.
3. Set the Banner type field:
   - Select **Fixed number of (user defined) steps** for manually defined steps.
   - Select **Generated based on panel history** for automatically generated steps. Automatically generated steps do not display initially but are added as the user reaches each step. Automatic banner text is generated using the Title field of each wizard panel.
   - Select **None** to disable the wizard banner.

Create a fixed wizard banner step
How to create fixed wizard banner steps.

Steps displayed in this banner can be manually defined or automatically generated.

1. Select **Fixed number of (user defined) steps** from the Banner type dropdown.
2. Select the Banner steps related list and click **New**.
3. Assign a name to your banner step. This name is what appears on the wizard.
4. Assign a display order (such as 100, 200, 300).
5. Repeat steps 2 – 4 for each banner step in the wizard.
6. On the wizard record, click the **Wizard Panels** related list.
7. In the Banner step field, enter the banner step that is completed when the wizard panel is displayed. Completed steps display in green in the wizard banner.

**Note:** You may need to **configure the list** to add the Banner step field.

---

Create a panel
How a user creates a panel.

1. Navigate to **System Wizards > Wizards**.
2. Open the wizard to which you are adding a panel.
3. In the Wizard Panels related list, click **New**.
4. Select the type of panel to create.
5. Enter basic panel information.

Table 164: Basic panel information

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Brief description of the panel</td>
</tr>
<tr>
<td>Expert</td>
<td>Name of the wizard</td>
</tr>
<tr>
<td>Title</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Next message</td>
<td>Label on the button that transitions to the next panel. This option does not apply for the last panel</td>
</tr>
<tr>
<td>Previous message</td>
<td>Label on the button that transitions to the previous panel. This option does not apply for the first panel</td>
</tr>
</tbody>
</table>

6. Enter type-specific information and save the record.
   - Prompts user to answer questions (wizard panel)
   - Catalog Checkout
   - Catalog Order
   - KB Viewer
   - Redirect
   - Record Generator
   - Survey (requires Survey Wizard Plugin)

**Edit a panel**

An Edit Panel button is available on wizard panels when testing the wizard. The button is available if the user can write to the expert record.

To control who can access this button, edit the write ACL on the Wizard [expert] table.

1. Navigate to System Wizards > Wizards.
2. Open the wizard containing the panel you want to edit.
3. Click Try It.
4. Click Edit Panel.
**Field setters**
A field setter defines a field value for a record created by a record generator.

Define field setters using:

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template</td>
<td>Name of the record generator panel.</td>
</tr>
<tr>
<td>Type</td>
<td>To define a static value (the same for each record created by the record generator), select <strong>Set field to a specific value</strong>. To define a value based on a wizard variable, select <strong>Set field to a variable</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Select the field name.</td>
</tr>
<tr>
<td>Value</td>
<td>Enter the value (static) or select the wizard variable name.</td>
</tr>
</tbody>
</table>

**KB viewers**
What a KB viewer panel displays.

A KB Viewer panel displays a knowledge base article. KB Viewer type-specific information is:
- KB Article - reference to the desired knowledge base article
- Title - label that appears above the article in wizard view

**Record generators**
A record generator panel creates a record in a table, such as an incident or change request.
There is no transition after the Record Generator. The following table includes fields specific to the Record Generator.

<table>
<thead>
<tr>
<th>Title</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>table in which to create a record (select Global when using a script)</td>
</tr>
<tr>
<td>Template</td>
<td>(optional) define field values on the new record using a template</td>
</tr>
<tr>
<td>Final view</td>
<td>(optional) enter the view in which to show the submitted form (such as ESS)</td>
</tr>
<tr>
<td>Script</td>
<td>script that runs when the panel is used. To learn more, see <strong>Wizard Scripts</strong>.</td>
</tr>
<tr>
<td>Field setters</td>
<td>define field values in the target table. To learn more, see <strong>Field Setters</strong>.</td>
</tr>
</tbody>
</table>

**Redirect panel**
A Redirect panel specifies a URL to which the user is taken upon transition to the panel.

Redirect type-specific information is:
• URL - URL location to which user should be taken
• Advanced - select this option to use a script
• Script - script that runs when the panel is used. To learn more, see *Wizard Scripts*.

*Use wizard banners*

The wizard banner is a graphical flow of wizard steps displayed at the top of a wizard.

![Wizard banner highlighted](image)

*Figure 212: Wizard banner highlighted*

*Activate a system wizard*

The System Wizards application is not active by default.

To activate system wizards:

1. Navigate to **System Definition > Application Menus**.
2. In the breadcrumbs, click **All** to display both active and inactive applications.
3. Search for **System Wizards**.
4. Ensure the Active field is set to true.

![Application panels](image)

*Define a basic transition*

How to define a transition.

---

**Note:** To learn more, see *Wizard Transitions*.

---

Define transitions:

1. In the Wizard Panels related list, click **Service Questions**.
2. In the Wizard Panel Transitions related list, click **New**.
3. Enter the transition details and click **Submit**.
4. Repeat steps 2 – 3 for all panel transitions in the table.

### Table 167: Panel transition details table

<table>
<thead>
<tr>
<th>Condition</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>incident is Email Issue</td>
<td>Service Questions</td>
<td>Email Incident</td>
</tr>
<tr>
<td>incident is Computer Issue</td>
<td>Service Questions</td>
<td>Computer Incident</td>
</tr>
<tr>
<td>incident is Password Reset</td>
<td>Service Questions</td>
<td>Password Incident</td>
</tr>
</tbody>
</table>

**Wizard concepts**

Parts of and concepts of a wizard. Wizards are built on the following concepts:
### Table 168: Wizards concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Panels</strong></td>
<td>Screens that appear in a defined order. Each step in a wizard is represented by a panel. The available panel types are:</td>
</tr>
<tr>
<td><strong>Table 169: Panel types</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Panel</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Wizard panel</td>
<td>prompts user to answer questions</td>
</tr>
<tr>
<td>Catalog checkout</td>
<td>displays an order confirmation screen</td>
</tr>
<tr>
<td>Catalog order</td>
<td>displays listings for service catalog items</td>
</tr>
<tr>
<td>KB viewer</td>
<td>displays a knowledge base article</td>
</tr>
<tr>
<td>Record generator</td>
<td>creates a record in a table</td>
</tr>
<tr>
<td>Survey</td>
<td>requires the Survey Wizard plugin</td>
</tr>
<tr>
<td><strong>Variables</strong></td>
<td>Questions that collect and store user input. Variables are defined in a wizard and can included on more than one panel.</td>
</tr>
<tr>
<td><strong>Transitions</strong></td>
<td>Define logic used to move between panels. Transition logic may be based on user input.</td>
</tr>
</tbody>
</table>

### Advanced customization

An advantage of wizards is the ability to implement advanced, custom functionality with a user-friendly interface.

Create advanced wizards using:

- **Wizard UI Policy and Client Scripts** - create dynamic effects and validation for wizards, panels, and variables.
- **Wizard Scripts** - run scripts from record generator panels, transitions, and UI policies.

### Create an advanced wizard (demo)

This example creates a wizard that associates assets and details and then creates a contract with this information.

The wizard uses a script to create a new record.

**Create the wizard**

How to create the wizard.

1. **Activate the system wizards application**, if necessary.
2. Navigate to **System Wizards > Wizards**.
3. Click **New**.
4. In the Name field, enter **Contract Creation**.
5. Right-click the header and select **Save**.

---

**Define a variable**
How to define a variable.

1. In the Wizard Variable related list, click **New**.
2. Enter the variable details and save the record.
3. Repeat steps 1 – 2 for all variables in the table.

---

**Table 170: Wizard Variables Details Table**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Question</th>
<th>Additional configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes/No</td>
<td>assets</td>
<td>Associate assets?</td>
<td></td>
</tr>
<tr>
<td>Select Box</td>
<td>contract_type</td>
<td>Which type of contract?</td>
<td>Mandatory: true Choice table: Contract [ast_contract] Choice field: Contract type</td>
</tr>
<tr>
<td>Date</td>
<td>starts</td>
<td>Start Date</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>ends</td>
<td>End date</td>
<td></td>
</tr>
<tr>
<td>Single Line Text</td>
<td>short_description</td>
<td>Enter short description for contract</td>
<td></td>
</tr>
<tr>
<td>List Collector</td>
<td>asset_select</td>
<td>Asset Listing</td>
<td>List Table: Configuration Item[cmdb_ci]</td>
</tr>
</tbody>
</table>
Create a panel demo
How to create the first two panels and add variables.

1. In the Wizard Panels related list, click New.
2. Select the panel Type of Prompts user to answer questions.
3. Enter the Name, then right-click the header and select Save.
4. In the Variables related list, click Edit....
5. Using the slushbucket, select and arrange the variables as listed in the table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Add variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Contract Screen</td>
<td>Associate assets?</td>
</tr>
<tr>
<td></td>
<td>Which type of contract?</td>
</tr>
<tr>
<td></td>
<td>Start Date</td>
</tr>
<tr>
<td></td>
<td>End Date</td>
</tr>
<tr>
<td></td>
<td>Enter short description for contract</td>
</tr>
<tr>
<td>2 Asset Screen</td>
<td>Asset Listing</td>
</tr>
</tbody>
</table>

Table 171: Panel variables table

Create the third panel and add a field setter
How to create the third panel and add field setters.

1. In the Wizard Panels related list, click New.
2. Select the panel Type of Creates something (record generator).
3. Enter the following information, then right-click the header and select Save.
   - Name: 3 Contract No Assets
   - Table: Contract [ast_contract]
4. In the Field Setters related list, click New.
5. Enter the field setter details and save the record.
6. Repeat steps 4 – 5 for all field setters in the table.

<table>
<thead>
<tr>
<th>Type</th>
<th>Field</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set field to a variable</td>
<td>Contract type</td>
<td>contract_type</td>
</tr>
<tr>
<td>Set field to a variable</td>
<td>Short description</td>
<td>short_description</td>
</tr>
<tr>
<td>Set field to a variable</td>
<td>Ends</td>
<td>ends</td>
</tr>
<tr>
<td>Set field to a variable</td>
<td>Starts</td>
<td>starts</td>
</tr>
</tbody>
</table>

Table 172: Field setter details table

Create the fourth panel and add a script
How to create the fourth panel and add a script.

1. In the Wizard Panels related list, click New.
2. Select the panel Type of Creates something (record generator).
3. Enter the following information, then right-click the header and select **Save**.
   - **Name**: 4 Create Contract and Assets
   - **Table**: Global

4. Paste the example script into the Script field. **Configure the form** to add the field, if necessary.

Example script:

```javascript
//Call the createContract function to obtain the ID of the created contract
var contract_id = createContract();

//Construct a url that will be used to redirect the user after submission
var uri =('ast_contract.do?sysparm_query=sys_id='+ contract_id);

//Redirect the user to the url
wizard.redirect= uri;
gs.addInfoMessage('Contract created');

//Create a new contract
function createContract(){
    var cc = new GlideRecord('ast_contract');
    cc.sys_class_name= wizard.contract_type;
    cc.starts= wizard.starts;
    cc.ends= wizard.ends;
    cc.short_description= wizard.short_description;
    var cntr = cc.insert();
    //We created a new contract and have the id so now we need the assets to associate
    associateAssets(cntr);
    return cntr;
}

//This is the way we loop through the elements in a list collector - assets in this case
function associateAssets(id){
    var items = wizard.asset_select.toString();
    items = items.split(',');
    for(var i =0; i < items.length; i++){
        var sys_id = items[i];
        if(sys_id !=''){
            var rl =new GlideRecord('ast_contract_instance');
            rl.ast_contract= id;
            rl.ci_item= sys_id;
            rl.insert();}
    }
}
```

*Define a transition (advanced demo)*

How to define a tradition.

1. Open a panel.
2. In the Wizard Panel Transitions related list, click **New**.
3. Enter the transition details and save the record.
4. Repeat steps 1 – 3 for all panel transitions in the table.
### Table 173: Panel transition details table

<table>
<thead>
<tr>
<th>Panel</th>
<th>Condition</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Contract Screen</td>
<td>assets is Yes</td>
<td>1 Contract Screen</td>
<td>2 Asset Screen</td>
</tr>
<tr>
<td>1 Contract Screen</td>
<td>assets is No</td>
<td>1 Contract Screen</td>
<td>3 Contract No Assets</td>
</tr>
<tr>
<td>2 Asset Screen</td>
<td></td>
<td>2 Asset Screen</td>
<td>4 Create Contract and Assets</td>
</tr>
</tbody>
</table>

*Test the wizard for a record with no assets*
Run through several scenarios to test transition logic and record generation.

1. Open the wizard record and click **Test**.
2. Enter the following information and click **Next**.
   - Associate assets?: No
   - Which type of contract?: select a value other than the default
   - Start Date: enter a date
   - End Date: enter a date
   - Enter short description for contract: enter a description
3. Verify that a new record is created and the values match step 2.

---

*Test the wizard for a record with an asset*
Test the wizard for a record with assets.

1. Open the wizard record and click **Test**.
2. Enter the following information and click **Next**.
- Associate assets?: Yes
- Which type of contract?: select a value other than the default
- Start Date: enter a date
- End Date: enter a date
- Enter short description for contract: enter a description

3. Verify that the asset screen opens.
4. Select assets to add to the list and click **Next**.
5. Verify that a new record is created and the values match steps 2 and 4.

---

**Publish the wizard demo**
Restrict access to users with asset role.

1. Open the wizard.
2. In the Roles field, select **asset** and save the record.

**Add the wizard as a module in the Asset Contracts application**
Add the wizard as a module in the Asset Contracts application.

1. Open the wizard.
2. Copy the sys_id.
3. In the navigation pane, right-click **Asset Contracts** and select **Edit Application**.
4. In the Modules related list, click **New**.
5. Enter the following information and save the record.
   - Title: Create Contract
List administration

Administrators control the data presented to end users in a list, the controls that appear, and the behavior of the controls.

Note: For information about using lists, see Lists on page 56.

List configuration

Users with the appropriate roles can configure various aspects of lists. Configuration changes apply to all users.

With list configuration, you can add, remove, and reorder list columns. You can configure calculations to appear under columns. You can also hide controls and define access conditions by role for existing list controls. Configuring lists is called personalizing lists in versions prior to the Fuji release.

Note: Configuring a list in this way modifies the list for all users. To make changes to a list that are visible to you only, see Personal lists on page 93.

Configure the list layout

You can configure a list to choose which columns appear in a list, create list views, and create new fields on the table.

Role required: personalize_list

1. Open the list in the view you wish to modify.
   For example, to modify the incident mobile view, navigate to Incident > Open and select View > Mobile from the context menu.
2. Right-click any column heading and select Configure > List Layout.
3. Use the slushbucket to select the columns and the order of appearance.
   The first non-reference field automatically links to the form view of the record. For this reason, consider using the record number as the first column in the list layout.
Note: If you create a personal list and then configure the list view differently, the changes do not appear until you reset your personal list to column defaults.

Add an extended field to a base table list
Administrators can configure a property that determines whether fields on extended tables can be added to a parent table list. For example, when this feature is enabled, you can view and filter on the Caller field, from the Incident table, on a Task table list.

Role required: admin

Note: Enabling this property does not show the extended table fields in the personalize list.

1. Navigate to System Properties > UI Properties.
2. Locate the property Allow base table lists (task, cmdb_ci, etc.) to include extended table fields (incident_state, os_version, etc.), and allow filtering on extended table fields (glide.ui.list.allow_extended_fields).
3. Select the check box to enable extended fields on parent table lists. Clear the check box to disable the feature.
4. Click Save.

Configure list calculations
You can configure list calculations for columns, which calculate column totals, minimums, maximums, and averages.

Role required: personalize_list

You can configure list calculations for multiple columns in the same list. List calculations apply only to the view of the list in which they are configured. All users can see configured list calculations.

Note: List calculations are available in list reports.

1. In a list of records, right-click the header of a numeric column you want to evaluate with list calculations.
2. Select Configure > List Calculations.
   A dialog box appears.

   ![Price (calculations)](image)

3. Select one or more of the following list calculation options:
   • Total value (not available for string, date, or date/time fields)
   • Minimum value
• Maximum value
• Average value (not available for string, date, or date/time fields)

4. Click **OK**.
   The calculations appear below the last record in the column. If the list is grouped, in addition to the overall calculations at the bottom of the list, group calculations appear below the last record in each group.

**Configure list controls**

You can configure list controls, such as buttons and filters.

Role required: personalize_control

1. Right-click a list column header.
2. Select **Configure > List Control**.
3. Complete the form.
Note: The list controls that are available for embedded lists are more limited. Unless otherwise noted, the list control fields in the following table are available for both standard and embedded lists.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>View the name of the table for the list or related list. For example, Change Request [change_request]. This value is set by the system automatically.</td>
</tr>
<tr>
<td>Related list</td>
<td>View the name of the table and field that define the related list. For example, sysapproval_approver.sysapproval. This value is set by the system automatically.</td>
</tr>
<tr>
<td>Label</td>
<td>Enter the label to display for this list. Allows an admin to customize the label for a related list or list. If not supplied, the default plural label for the file is used. For example, the label for the Incident table would be Incidents.</td>
</tr>
<tr>
<td>Omit new button</td>
<td>Select the check box to prevent the New button from displaying on this list. Clear the check box to display the New button or to control the New button with roles (New roles field). This field is available for standard lists only.</td>
</tr>
<tr>
<td>Omit edit button</td>
<td>Select the check box to prevent the Edit button from displaying on this list. Clear this check box to display the Edit button or to control the Edit button with roles (New roles field). The Edit button does not apply to all lists. This field is available for standard lists only.</td>
</tr>
<tr>
<td>Omit if empty</td>
<td>Select the check box to omit the Related List from the form entirely (no header) if there are no entries for the Related List.</td>
</tr>
<tr>
<td>Omit columns if empty</td>
<td>Select the check box for a top-level list to omit the column headers AND filters/breadcrumbs for an empty column.</td>
</tr>
<tr>
<td>Omit filters</td>
<td>Select the check box to hide filters or breadcrumbs for this list. Clear this check box if you always want filters or breadcrumbs or to control filters/breadcrumbs with roles (Filter roles field). This field is available for standard lists only.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Omit links</td>
<td>Select the check box to hide links for fields that reference other files in this list. Leave this button unchecked to generate links or to control the use of links with roles (Link roles field).</td>
</tr>
<tr>
<td>Omit drill-down link</td>
<td>Select the check box to disable the link to the record from the first column in list view. Users can still click the reference icon to access the record. This field is available for standard lists only.</td>
</tr>
<tr>
<td>Hierarchical lists</td>
<td>Inserts a hierarchical list into a record list. Hierarchical lists enable a user to view the contents of a record's related lists without leaving the record list form. This field is available for standard lists only.</td>
</tr>
<tr>
<td>List edit type</td>
<td>Controls the ability of a user to edit values directly in individual cells in a list. The options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Save immediately (cell edit mode):</strong> enables cell editing. The entire row is saved when the user enters a new value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Save data by rows:</strong> enables cell editing. The row is saved only when the user navigates away from the row or clicks the <strong>Save</strong> icon (✔️). This mode allows the user to modify multiple values before saving a record.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Disable list editing:</strong> prevents users from editing cells in the list.</td>
</tr>
<tr>
<td>List edit tag</td>
<td>Enter an arbitrary string of letters or numbers to create a unique tag that is sent to a reference qualifier as the script variable listEditRefQualTag.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>List edit insert row</td>
<td>Select the check box to enable or clear the check box to disable the ability for a user to create new records in list view. When it is enabled, an empty row appears at the bottom of the list.</td>
</tr>
</tbody>
</table>

This field is available for standard lists only.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New roles</td>
<td>Specify the user roles that can access the New button on this list. Leave the field blank to enable all users to access the New button. This option does not apply to embedded lists, which do not contain New buttons.</td>
</tr>
</tbody>
</table>

| Edit roles | Specify the user roles required to have the Edit button appear in the list. This field is available for standard lists only.                                                                                   |
| Filter roles | Specify the user roles required to have the filter appear in the list. This field is available for standard lists only.                                                                                     |
| Link roles | Specify the user roles required to have links generated for fields that reference other files.                                                                                                              |

**Example of configuring list controls**

The following example shows how to configure a list of related incidents in a problem record to suit specific business needs.

Role required: personalize_control

In this example, a related list label is configured to say Child Incidents, not just Incidents. The related list is also configured to remove the New button to prevent users from creating new incidents from the Problem form.

1. Navigate to a problem record.
2. Right-click in a column header of the embedded Incidents list and select Configure > List Control.
3. On the List Control form, change the Label field to Child Incidents.
4. Select the Omit new button check box.
5. Click **Submit**.
In the problem record, the label for the embedded incident list has changed and the **New** button is no longer available.

### Advanced list control with scripts

You can write scripts to specify which list elements, such as links, filters, and buttons, are present.

These scripts can react to the record that is being displayed. For related list controls, the parent record is identified by the global variable `parent`. For primary lists, there is no parent record.

**Configure** the List Control form and add the script field that you need:

- **Omit Columns Condition**: Omit column headings if there are no conditions.
- **Omit Edit Condition**: Omit the **Edit** button.
- **Omit Empty Condition**: Omit the list header if there are no conditions.
- **Omit Filter Condition**: Omit the list filter.
- **Omit Links Condition**: Omit related links.
- **Omit New Condition**: Omit the **New** button.

In these script fields, if the script evaluates to `true`, the item is omitted. The following script is an example that you can use on the **Affected CIs** related list to show only the **Edit** button if the parent task is active. This script is placed in the **Omit Edit Condition** script field.

```javascript
var answer;
if (parent.active == true) {
    // Do not remove the 'Edit' button
    answer = false;
} else {
    // Remove the 'Edit' button
    answer = true;
}
answer;
```

**Configure a list UI action**

You can create a UI action that opens the slushbucket interface to customize lists in the list view

Role required: **ui_action_admin**

1. Navigate to **System UI > UI Actions**.
2. Click **New**.
3. In the **Table** field, select the **List [sys_ui_list]** table.
4. Select the **List context menu** check box.
5. Ensure the **Active** check box is selected.
6. Paste the following script in the **Script** field:

```javascript
var url = "slushbucket.do?sysparm_form=list&sysparm_list=" + current.
    name + "&sysparm_view=" + current. view. name;
url += "&sysparm_collection=" + current. parent + 
    "&sysparm_collection_relationship=" + current. relationship;
url += "&sysparm_referring_url=sys_ui_view.do?sys_id=" + current. view.
    toString () ;
action. setRedirectURL (url ) ;
```
7. Click **Submit**.
Controlling the sort sequence used to display lists

All lists have a default sort sequence based on the type of fields present in the list.

When a user displays a list for the first time, it is sorted by one of the following items.

- The **order** field if one is present in the table
- The **number** field if one is present in the table
- The **name** field if one is present in the table
- The field specified as the display field for the table

Setting the default sort order in the system dictionary

An administrator can set the sort order of records displayed for a table in the system dictionary when there is no other sort specified. This sort is the sort order presented to all users initially. Once a user sorts a list, that user preference is saved, and the list is always sorted for that user based on the previous sorting preference.

Sort order control

For any list, if the user has clicked a column header to sort by that column, then that action is remembered. The next time that list is displayed, the same field is used to order the list.

This remembered sort field can be overridden by adding a `sysparm_order` specification to the definition of the module. For example, if each time **News** is displayed, you want the new items listed in order of importance, then the **News** module can be updated accordingly.

```
&sysparm_order=sysparm_order=importance
```

**Figure 213: Set module sort order to the Importance field value**

The actual order (ascending or descending) can be specified by adding a `sysparm_order_direction` specification. Here are two examples of sorting a list by the `sys_created_on` field, one ascending and one descending:

```
&sysparm_order=sys_created_on &sysparm_order_direction=desc
&sysparm_order=sys_created_on &sysparm_order_direction=asc
```

Sort by multiple fields

To sort a list by multiple fields, remove everything from the filter field and place it in the arguments field. For example, if the filter specifies `[Active] [is] [true]` and you want to sort by category and subcategory, remove the condition and put the following in the module arguments:

```
active=true^ORDERBYcategory^ORDERBYsubcategory
```

This argument first orders by the category and then orders by the subcategory. If you want any field to be a descending sort then change `ORDERBY` to be `ORDERBYDESC`.  

© 2017 ServiceNow. All rights reserved.  680
List editor administration

The list editor allows users to edit field values directly from a list without navigating to a form.

Administrators can manage this feature by using the following options.

- Configure global properties
- Configure list control settings for a table
- Configure contextual security rules
- Manage user preferences

**Warning:** Client scripts and UI policies run on forms only (client-side) and do not apply to list editing. Allowing list editing with client scripts running on fields in a form can result in incorrect data being saved to the record. For systems in which client scripts or UI policies apply to forms, either disable list editing or create appropriate business rules or access control to control the setting of values in the list editor. See [Use business rules and client scripts to control field values](#) on page 3811 for information on managing form and list editing.

Configure list editor properties

You can configure list editor properties that control whether lists can be edited, and which field types cannot be edited.

Role required: admin

1. Navigate to **System Properties > UI Properties**.
2. To disable list editing, set the **Enable list editing** (glide.ui.list_edit) property to **No** by clearing the check box.
   
   This property is enabled by default, and it globally enables list editing. When you disable it, the list editor is disabled globally.

3. To configure the field types that cannot be edited, complete the following steps.
   a) Locate the **List of element types (comma-separated) that cannot be edited in the list editor** (glide.ui.list_edit_ignore_types) property. It contains several element types that cannot be edited by default.

   The following field types are not editable from the list editor by default.
   
   - Conditions [conditions]
   - Currency [currency]
   - Document ID [document_id]
   - Field List [field_list]
   - HTML [html]
   - Image [user_image]
   - List [glide_list]
   - Price [price]
   - Template Value [template_value]
   - Time [glide_time]
   - User Roles [user_roles]
   - Video [video]

   b) Add any other field types you want to disable to the end of the list, separated by a comma.

4. Click **Save**.
Configure list control settings for the list editor

You can configure the list control settings that affect the list editor.

Role required: personalize_control

List control settings customize the behavior of list functions for a table.

1. Navigate to a list view for the desired table.
2. Right-click a list column header and select Configure > List Control.
3. Select the desired settings.

Table 175: Control settings for the list editor

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List edit type</td>
<td>Controls the ability of a user to edit values directly in individual cells in a list. The options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Save immediately (cell edit mode)</strong> enables cell editing. The entire row is saved as soon as the user enters a new value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Save data by rows</strong> enables cell editing. The row is saved only when the user navigates away from the row or clicks the</td>
</tr>
<tr>
<td></td>
<td><strong>Save</strong> button. This mode allows the user to modify multiple values before saving a record.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Disable list editing</strong> prevents users from editing cells in the list.</td>
</tr>
</tbody>
</table>

| List edit insert row | Controls the ability of a user to create new records in list view. Select the check box to enable or clear the check box to   |
|                     | disable this ability. When it is enabled, an empty row appears at the bottom of the list.                                      |

4. Click **Update**.
Managing user preferences for list editing

Users can set user preferences for list editing by personalizing a list (available when the list mechanic is activated).

To view and manage list editor user preferences, navigate to User Administration > User Preferences.

Table 176: User Preferences

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_edit_enable</td>
<td>Set the value to true to enable or false to disable list editing. Default value is true.</td>
</tr>
<tr>
<td>list_edit_double</td>
<td>Set the value to true for double-click or false for single-click. Default value is true.</td>
</tr>
</tbody>
</table>

Configuring contextual security for the list editor

The list editor enforces existing access control rules (ACLs) and additional security controls to restrict editing from a list.

The list_edit security operation specifically controls the ability to edit information in a list. Apply this operation in the same manner as the write operation to limit list editing for fields that require the user to navigate to the form. Reasons you may require a user to edit a field in a form include complex UI policy constraints or other policies you have in place.

The list_edit security operation specifically controls the ability to edit information in a list. Apply this operation in the same manner as the write operation to limit list editing for fields that require the user to navigate to the form. Reasons you may require a user to edit a field in a form include complex UI policy constraints or other policies you have in place.

• write and list_edit access to the field
• write and list_edit access to any dependent fields of the field
• write and list_edit access to any fields that are dependent on the field being edited

To configure access controls, navigate to System Security > Access Controls. The following examples use the list_edit security operation to restrict list editing in certain contexts.

Example - Restrict a table

This access control prevents everyone from editing all fields in the Incident table in a list.
Figure 214: Restrict the Incident Table

- **Type**: record
- **Operation**: list_edit
- **Name Incident**: [incident]
- **Admin overrides**: Clear the check box.
- **Script**: `answer = false;`

Example - Restrict a field

This access control prevents everyone except an administrator from editing the Short Description field of an incident record in a list.
Figure 215: Restrict the Incident Short Description

- **Type**: record
- **Operation**: list_edit
- **Name**: Incident [incident], Short Description
- **Admin overrides**: Select the check box.
- **Script**: `answer = false;`

**Example - Restrict a field with a script**

This access control prevents everyone from editing an incident with a category of Software in a list. It is defined by a script.
**Figure 216: Restrict Software Incidents**

- **Type:** record
- **Operation:** list_edit
- **Name Incident:** [incident]
- **Admin overrides:** Clear the check box.
- **Script:**

  ```java
  if (current.category == 'software')
  answer = false;
  else
  answer = true;
  ```

**Example - Restrict a field with a condition**

This access control prevents everyone from editing a Critical Incident in a list. It is defined by a condition.
Figure 217: Restrict Critical Incidents

- **Type**: record
- **Operation**: list_edit
- **Name Incident**: [incident]
- **Admin overrides**: Clear the check box.
- **Condition**: Priority is not 1 - Critical

Personal list administration

Users can customize the layout for any list view by personalizing a list. Administrators can manage the personal lists of other users.

Administrators can manage personal lists using the following options:

- Activate or deactivate it globally
- Control user access to it based on roles
- Manage users’ personal lists

The personal list function requires the list mechanic UI macro.
Enable or disable personal lists

You can enable or disable personal lists through the UI Macros module.

Role required: ui_macro_admin

1. Navigate to **System UI > UI Macros**.
2. Activate (or deactivate) the **list_mechanic2** UI macro.

Control which roles can personalize lists

You can control who can create personal lists.

Role required: admin

1. Navigate to **System Properties > UI Properties**.
2. Locate the property **List of roles (comma-separated) that can use Personal Lists. If blank, all can use** (glide.ui.list_mechanic.roles).
3. Enter the roles for which personal lists are available, separated by commas, or leave the field blank to allow access for all users (default).

Manage personal lists

Administrators can manage personal lists.

Role required: admin

1. Navigate to **System UI > Lists**.
2. Review personal lists that have a value in the **User** field.

Administer detail rows

Detail rows provide additional information about records in a list.

Role required: admin

To display detail rows, an administrator must enable them globally and add them to lists.

**Note:** Detail rows are supported in UI15 and up.

1. To enable detail rows, *add a new system property* with the following field values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.ui.list.detail_row</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

2. To add detail rows to a list, complete the following steps.
   a) Add the **detail_row** *dictionary attribute* to the table.
      Specify the name of the field to display in the following format:
      detail_row=short_description
b) Use the all_tables element descriptor with the detail_row attribute to display detail rows for all child tables that contain the designated field.
For example, all_tables.detail_row=short_description is on the Task [task] table by default, so the short description detail row appears in lists for Incident [incident], Problem [problem], and other child tables unless you set detail_row to a different field for each one.

Restrict filters and breadcrumbs with fixed queries

The record list view allows users to navigate to different subsets of a table using breadcrumbs and filters. You can limit access to parts of the table by restricting active links in breadcrumbs or by suppressing breadcrumbs and filters for specific roles.

Role required: admin

A breadcrumb option enables an administrator to control the base view of a record list presented to users. By adding a fixed query to the argument for a module, an administrator can prevent users from expanding their view past a specified starting point. The argument for this fixed query is written as &sysparm_fixed_query=active=true. A use case for this query is to prevent users from using the breadcrumbs to switch a list of open incidents to a list of all incidents. When users select Incident > Open, they are limited to viewing and filtering a list of open (active=true) incidents.

1. Perform the appropriate action for your version of the UI:

| UI16          | 1. Navigate to System Definition > Application Menus.  
               | 2. Open the application menu that contains the module you want to edit. |
|---------------|-----------------------------------------------------------|
| UI15 or UI11  | Right-click the application in the navigation pane and select Edit Application. |

2. Select the module to edit.
   For example, select Open.

3. In the Link Type section, select List of Records for the Link type.

4. Delete the Active is true filter, if present.

5. Add &sysparm_fixed_query=active=true to the Arguments field and update the record.
6. Navigate to Incidents > Open and examine the breadcrumbs.

The Active=true link is the widest view permitted in the list of Open incidents. Notice that the breadcrumb for All is not linked.
Suppress filters and breadcrumbs with list controls

You can use list controls to suppress filters and breadcrumbs on defined tables for users with specific roles.

Role required: personalize_control

1. Navigate to the record list view of the table to restrict breadcrumbs.
2. Right-click any column header, and select Configure > List Control.
   You can also navigate to System UI > List Control.
3. Select the Omit Filters check box. Clear the check box to include filters.

The Filter Roles field allows an administrator to specify certain roles who can see the filters.

Suppress filters and breadcrumbs using script includes

You can use a script to restrict filters and breadcrumbs to specific roles, either on a per-table basis or global basis. Using a script is an advanced option that offers additional flexibility compared to using list control.

Using a script include requires knowledge of JavaScript.

Role required: script_include_admin or admin

1. Create a script include with the name <tablename>DisplayFilter.
   The script section contains one function with the same name as the script include.
2. Use your function to set the global variable answer to either true (show the filters and breadcrumbs) or false (hide them.)
   The following example restricts filters and breadcrumbs on the Incident table to users with any role. Be sure that the name of the script matches the function name exactly, including case.

   ```javascript
   var incidentDisplayFilter = Class.create();
   ```
incidentDisplayFilter.prototype = {
    type: 'incidentDisplayFilter',
    initialize: function() {
    },
    incidentDisplayFilter: function() {
        var answer;
        if (gs.hasRole()) {
            answer = true;
        } else {
            answer = false;
        }
        return answer;
    }
};

3. Optional: To exclude a specific role from having access to filters and breadcrumbs, make the following change.

var incidentDisplayFilter = Class.create();
incidentDisplayFilter.prototype = {
    type: 'incidentDisplayFilter',
    initialize: function() {
    },
    incidentDisplayFilter: function() {
        var answer;
        if (gs.hasRole() && !gs.hasRole('newrole')) {
            answer = true;
        } else {
            answer = false;
        }
        return answer;
    }
};

Users with the role newRole cannot access filters and breadcrumbs.

4. Optional: To let all users use filters and breadcrumbs on the Incident table, make the following change to your script.

var incidentDisplayFilter = Class.create();
incidentDisplayFilter.prototype = {
    type: 'incidentDisplayFilter',
    initialize: function() {
    },
    incidentDisplayFilter: function() {
        var answer = true;
        return answer;
    }
};
5. Optional: To modify filter and breadcrumb access for another table, create a script include using the name of that table instead of Incident.

Increase the allowed number of breadcrumb entries

You can add a property to allow for a larger number of breadcrumb entries in the filter.

Role required: admin

1. Enter `sys_properties.list` in the Navigation filter.
   The entire list of properties in the System Properties [sys_properties] table appears.

2. Verify the property does not already exist by searching for `glide.ui.breadcrumb_max_entries`.

3. Click **New**.

4. Complete the form as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.ui.breadcrumb_max_entries</td>
</tr>
<tr>
<td>Type</td>
<td>integer</td>
</tr>
<tr>
<td>Value</td>
<td>The number of breadcrumb entries you want to appear in the filter, for example, 15. The default number is 10.</td>
</tr>
</tbody>
</table>

5. To verify this property, go to any table and use the filter modifier **is one of** to search for any number of items. The number of entries you entered in the Value field will display before ending in a […].

![Figure 218: Breadcrumb entries](image)

Enable a hierarchical list

You can enable hierarchical lists to make data from related lists directly accessible from within a record list.

Role required: personalize_control

Hierarchical lists enable a user to view the contents of a record's related lists without leaving the record list form.

---

**Note:** You cannot enable a hierarchical list view for an embedded list.

1. Right-click any column header, and select **Configure > List Control** from the context menu.
2. Select the **Hierarchical lists** check box in the List Control dialog box and then click **Submit** or **Update**.
3. Click the arrow for a record number to display the related lists for the record.
Use list controls in hierarchical lists

By default, no related list tabs are visible in a hierarchical list and only a single related list is displayed at a time. The column headers are hidden by default.

Role required: none

1. To select a different related list, click the link in the list title.
2. Select a list from the menu.
If the related list is empty, the hierarchical list indicates that there are no records to display.

The platform remembers this preference and displays the last selected related list for each table when you open another hierarchical list within that table.

3. To display column headers in the related list, click the show/hide headers icon in the upper right corner of the hierarchical list.
The platform remembers this preference and opens all hierarchical lists with column headers enabled until the headers are hidden.

Context ranking

Context ranking allows a user to sort a collection of records preferentially, that is, independently of the attributes of those records.

For example, a Customer Support manager can organize a list of incidents in the order in which a technician should work on the tasks. Creating such an arbitrary list with a list filter is not possible.

Activate context ranking

You can activate the Context Ranking plugin (com.glide.sorting) if you have the admin role. This plugin activates related plugins if they are not already active.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

## Create a ranking definition

Enable context ranking for a related list by creating a ranking definition.

**Role required:** admin

1. Navigate to **System Definition > Ranking Definitions**.
2. Click **New**.
3. Enter a **Name** for the ranking definition.
4. In the **Record table** field, select the table for which ranking is enabled.
   This is the table of the records in the ranked related list. For example, to enable ranking for a related list of incidents in the Problem form, select the Incident [incident] table.
5. In the **Context column** field, select the reference column providing the context in which ranking is to be performed.
   In this case, the context is the form in which the related list appears and not a table name. For example, a related list with a **Record table** of Incident [incident] can have a number of contexts, including the Problem form or a **Configuration item** form.
6. Complete the relationship-based fields if the ranking is to be performed on a related list that is defined by a relationship.
   - **Relationship table**: select the table forming the other side of the relationship. In the case of the ranking definition for Product stories in scrum, you might use the Product Model [cmdb_model] table.
   - **Relationship column**: select the column from the relationship table to compare with the context column. To avoid adding a related list of user stories to the Product Model [cmdb_model] table, you might use the Sys ID column, which defines a subclass of the Product Model table (cmdb_model.sys_id).
7. In the **Attributes** field, enter attributes to change and restrict the contents of the Rank dialog box, as appropriate.
   - **visible_columns**: a semi-colon separated list of columns to be displayed in the Rank dialog box.
     If not specified, the Rank dialog box uses the default view of the related list you selected in the **Record table** field.
   - **extra_conditions**: an encoded query to restrict the records shown in the Rank dialog box. For example, the query `extra_conditions=^state!=7` displays all incidents whose state is not Closed. If not specified, the Rank dialog box shows all records for the relationship.
8. Click **Submit**.

The index is generated automatically. The system completes these read-only fields:

- **Index column**
- **Rank ui action**
- **Context menu item**

**Note:** After you submit a ranking definition, only the **Attributes** field can be edited. If additional changes are required, delete the record and create a new one with the revised settings.

The system does not generate indexes for ranking definitions inserted into the database by an update set. To use a ranking definition inserted by an update set, you must generate the indexes manually. To generate indexes, open the Ranking Definition form and click **Generate Indexes**.
Rank a list

After you create a ranking definition for a table, related lists based on that table and context include options for ranking the list and displaying the list by rank.

Role required: none

You can set the preferential order of records.

1. Click the **Rank** button in the related list to reorder the records.
The Rank dialog box appears, allowing you to sort the records in any order.
2. Click and drag the move icon ( ) to change a record's position.

**Note:** Administrators can define which columns appear in the Rank dialog box in the Attributes field of the Ranking Definition form.

**Apply a new sort order to a list**
You can apply a new sort order to the records in a list.

Role required: none

1. Open the context menu from the list title and select **Sort by rank.**
2. To remove the ranking and return to the original order of the list, click the label in any column that contains data.

Scrum ranking definitions

Ranking definitions for user stories are part of the SDLC (scrum process) application. These ranking definitions enable scrum users with the proper roles to rank related lists of stories in the these tables:

- Product stories: Application Model [cmdb_application_product_model] table
- Release stories: Release [rm_release] table
- Sprint stories: Sprint [rm_sprint] table

Users can rank stories in the scrum planning board by priority. Story lists ranked in the planning board appear in the new order in product, release, and sprint forms. Stories ranked in one of these scrum forms changes the order in the planning board.

Story points for each story are listed in a ranked view, which is useful for prioritizing stories.
The Context Ranking feature enables all scrum users to manually sort a related list of stories by priority. A product owner or release planner uses this tool to establish the order in which the stories are to be worked. Stories ranked in a related list (in the Product, Release, and Sprint forms) appear in the same order in the planning board when the appropriate backlog is displayed. Conversely, stories ranked in the planning board appear in that order when viewed in the related list in the form. Users can switch the view in a Stories related list from ranked to any sort order without changing the ranking in the planning board.

1. In the Stories related list, click Rank.
   A dialog box appears, allowing you to arrange the stories in any order, such as by priority.
2. Click and drag each story into position using the move icon to the left of the story number.
3. When you are finished, click the X in the upper right corner to close the list.
   The Stories related list is not sorted by your ranking initially.
4. To view the related list in its ranked order, open the context menu from the related list and select Sort by rank.
5. To return the sort order of the list to an unranked state, click once in the heading of any column that contains data.

The system uses this ranked list to display the appropriate backlog in the planning board. For example, if you rank stories in the Stories related list in the Release form, the release backlog in the planning board uses the same ranking to display the stories. Conversely, scrum masters, product owners, and release planners can create a new ranking order for the Stories related list by rearranging the list of stories in the planning board.

Form administration

Configure forms to add new content, create custom forms, and personalize form layout.

Configure a form

The layout of a form can be configured for any form view.

Role required: personalize_form

1. Click the form context menu icon and select Configure > Form Layout.
2. Using the slushbucket, select the fields and the order in which you want them to appear.
   Available items that appear in green followed by a plus (+) sign represent related tables. To access fields on these tables, use dot-walking.
3. Click **Save**.

**Warning:** It is not recommended to add the same field to more than one section of a form unless the field displays read-only data. Having two or more instances of an editable field can cause data loss and prevent the proper functioning of UI and data policies.
Configure task record insert options

The Insert and Insert and Stay options are disabled by default for task records such as incidents and change requests. You can set a system property to show these options for task records.

Role required: admin

1. Navigate to System Properties > UI Properties.
2. Locate the property Allow the use of the "Insert" and "Insert and Stay" options on task derived tables. (glide.ui.task.insert)
3. Select the check box to enable or clear the check box to disable (default) the options for tasks.
4. Click Save.

Add a related list

You can configure related lists to appear on forms and in hierarchical lists.

Role required: personalize_form

Related lists display records in another table that have a relationship with the current record.

1. Open the form.
2. Click the form context menu icon and select Configure > Related Lists.
3. Using the slushbucket, select the related list to display on the form.
4. Click Save.
   Related lists appear at the bottom of the form.

Add a chart to a form

You can add a chart to a form.

Role required: personalize_form

The following report types are not supported on forms: List, Pivot, Multilevel Pivot, Calendar, and Single Score.
Figure 220: Chart on a form

1. Open the form.
2. Click the form context menu icon and select **Configure > Form Layout**.
3. Using the slushbucket, select *Chart*.
4. Enter a **Label** in the chart details.
5. Click **Save** to return to the form.
6. Click **Configure chart**.
7. Select a chart in the **Report** field and configure other options as desired.
8. Click **Update**.

Configure customer updates indicator

Customer updates are changes that are tracked by update sets, such as scripts, service catalog items, and other configuration tables.

Role required: admin

The customer updates indicator icon () may appear on the header of forms that have customer updates. Clicking the customer updates indicator opens the update set records for the item.
You can configure this indicator to appear for all or for specific administrators using the owned_by_indicator.form user preference.

1. Navigate to User Administration > User Preferences.
2. Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable the indicator for all administrators</td>
<td>Set the owned_by_indicator.form preference to true.</td>
</tr>
</tbody>
</table>
| Enable the indicator for an individual administrator | 1. Set the owned_by_indicator.form preference to false.  
2. Create a new user preference with the following values:  
  • Name: owned_by_indicator.form  
  • User: Select the administrator for which to enable the preference.  
  • Value: true |

Form design

Administrators or users with the personalize_form role can use the form design feature to quickly create new or change existing form views.

Form design is an alternative to configuring forms that combines several configuration options into one tool.

Form design is available by default for new instances and requires UI16 or UI15. Administrators who upgrade from a previous version must activate the Form Designer (com.glide.ui.ng.fd) plugin to use the feature.

This video highlights the features of Form Design to create and customize forms.

Parts of the form design interface

The form design interface is divided into sections.
**Page header**

The page header displays the current table and view being designed. Each form is tied to a unique table and view combination. For example, there can only be one form for the incident table default view. Select a different view to modify different form layouts for a table.
Figure 222: Header

Field navigator

The field navigator allows you to manage fields on the form. It contains the following elements:

- **Filter:** Allows text search on the currently selected tab.
- **Fields:** Displays a list of existing fields you can add to the current form.
- **Field Types:** Displays a list of field types you can add to the current form. Adding a field type to the form layout creates new fields in the selected table when the form layout is saved.
Field properties are accessed by clicking the gear icon that appears when you point to a field in the form layout.

Figure 223: Field navigator
The form layout displays a separate cell for each field, section, annotation, UI element, and related list that is on the form. Each cell has an identifying label and a handle icon to aid in selecting and moving it. The location of the cell in the form layout represents its relative position on the form. Only fields visible on the form layout are displayed on a form.
Figure 225: Form layout
Change or create new form
You can create a form or change the content of a form.
Role required: personalize_form
1. Navigate to the form.
2. Right-click the form header and select Configure > Form Design. The form design interface opens in a new browser tab or window.
3. Optional: At the top of the Form Design:
   a) Select the table for the form.
   b) Select a form view to change, or click New to create a form for the selected table.

   By default, form design displays the form layout for the currently selected table and view.
4. Modify the form as desired.
5. Click Save.

Select which field to display
Move fields to and from the form workspace to display the fields on the form.
Role required: personalize_form

   Note: Use the filter to quickly find fields in the field navigator.

1. Navigate to the form designer.
2. From the list of fields in the Field Navigator, select the field you want to display from the list of available fields.
3. Drag the field to the form layout.
4. Drag the field to the location you want it to be on the form.
5. To remove a field from the form layout, select the field and click the Remove this field (X) button beside the field label.
6. Click Save.

Add a new field
You can add a field to a form and the associated table.
Role required: personalize_form

Adding a field creates a dictionary entry for the new column in the current table. You can create a custom table and then use the form designer to add fields to the form.

   Note: Use the filter to quickly find field types in the field navigator.

1. Navigate to the form designer.
2. Click the Field Types tab.
3. Select the cell for the field type to create.
4. Drag the field type to the location you want it to be on the form.
5. Enter a label for the field.
6. Optional: Point to a field in the form design area and click the gear icon to access properties.
7. Click **Save**.

### Add a new section

Move sections to and from the form layout to display or remove the sections on the form.

Role required: personalize_form

Sections can have a one- or two-column layout.

**Note:** By default, the first section on a form is always a read-only section displaying the label of the table. You cannot remove this section.

1. Navigate to the form designer.
2. Select an existing section.
3. Click the (+) button beside the section label to add a section.
4. Enter a label for the section.
5. Drag sections to reorder them.

**Note:** The label for the first section on the form designer is displayed as the form title.

6. To delete a section, click the (x) button beside the section label.
7. Click **Save**.

### Add an annotation

Use annotations to provide instructions or additional information about fields on a form.

Role required: personalize_form

1. Navigate to the form designer.
2. Click the **Field Types** tab.
3. Select the **Annotations** field type.
4. Drag the annotation to the appropriate location on the form.
5. Point to the **Annotations** field and click the gear icon to open Properties dialog box.
6. In the Properties dialog box, enter a value for **Annotation Text**.
   You can use HTML tags to format the annotation text.
7. Click **Save**.

### Add a formatter

Use formatters to add information from macros or previously scripted UI elements to a form.

Formatters are used to add information from macros or previously scripted UI elements to a form.

1. Navigate to the form designer.
2. Click the **Fields** tab.
3. In the filter, enter the string **Format**.
4. Select the formatter you want to add to the form.
5. Drag the formatter to the location you want it to be on the form.
6. Click **Save**.
Form personalization administration

Users can customize the layout for any form view by personalizing the form. Form personalization is available in UI16 and UI15.

Administrators can manage this function using the following options.

- Activate or deactivate form personalization globally
- Control user access to form personalization based on roles
- Manage users’ personalized forms

**Note:** Personalizing a form in this way modifies the form for you only. To make changes to a form that are visible to all users, you must configure the form.

Activate form personalization

Form personalization is activated for new instances. To activate form personalization for upgraded instances, an administrator must activate the Form Personalization (com.glide.ui.personalize_form) plugin.

Role required: admin

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Change form personalization role requirements

By default, the itil role is required to personalize forms. You can change the role requirements.

Role required: admin

1. Enter **sys_properties.list** in the navigation filter.
2. Locate the glide.ui.personalize_form.role property in the System Properties list.
3. In the **Value** field, specify the roles that can access form personalization.

Manage personalized forms

When a user personalizes a form, the system stores the customizations as a user preference record. You can view and manage the user preferences.

Role required: admin

1. Navigate to **User Administration > User Preferences**.
2. Filter the list by **[Name] [contains] [personalize]**.
   - There is a user preference for each form view each user personalizes. The name format combines the word personalize with the name of the table and the name of the view. For example, if a user personalizes the default view of the Asset [alm_asset] form, the user preference is called personalize_alm_asset_default.
3. Delete a user preference to remove the customizations for the user.

**Disable form personalization**

You can disable form personalization.

Role required: admin

Activating the Personalize Forms plugin sets the glide.ui.personalize_form property to true. You can disable form personalization.

1. Enter `sys_properties.list` in the navigation filter.
2. Locate the `glide.ui.personalize_form` property in the System Properties list.
3. Set the `Value` field to `false`. The personalize form icon is removed from forms.

**Form annotation**

A form annotation is additional information on a form, such as a line or paragraph of text. Use form annotations to provide on-screen instructions to your users.

**Toggle annotations**

Users can toggle annotations on a form in UI16 and UI15.

The visibility of annotations for each user is controlled by the `glide.ui.show_annotations` user preference.

1. Navigate to a form.
2. Perform the appropriate action for your version of the UI.

<table>
<thead>
<tr>
<th>UI16</th>
<th>Click the more options icon (●●●) to see the annotations icon. The icon is light grey and cannot be toggled when no annotations are available.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI15</td>
<td>Click the annotation toggle icon (?) to show or hide annotations.</td>
</tr>
</tbody>
</table>

**Add a form annotation**

You can add annotations to forms.

Role required: personalize_form

1. Navigate to a form.
2. Right-click the header and select **Configure > Form Layout**.
3. Under **Form view and section**, select a section.
4. In the Available list, double-click *Annotation* to move it to the **Selected** list. The Annotation Details section appears.

© 2017 ServiceNow. All rights reserved. 716
5. Place it above the field to annotate.
6. In Annotation Details, select the type of annotation. The available types of annotations include **Info Box Blue**, **Info Box Red**, **Line Separator** (any text you add appears beneath the line), **Section Details**, **Section Separator**, and **Text**.
7. Select either Plain Text or HTML for the data type and enter the annotation in the text box. All valid HTML tags are supported.

   For example, select the Text annotation type, the HTML data type, and enter `<span style="color:red">Select the primary location:</span>`. The text appears red on the form.

8. Click Save.

Support multiple languages for a form annotation

You can store multiple translations of form annotation text.

Role required: admin

To support multiple languages, use message records to translate annotation text.

1. Navigate to System UI > Messages.
2. Create a message record for each language you support.
3. On the Message form, set the **Key** field to a unique identifier for the annotation text.
   The annotation text is a good key. The key must be the same for each translation message for the annotation.

4. Select the appropriate **Language**.

5. In the **Message** field, enter the translated annotation text.

6. Edit the form annotation and reference the message key with a `gs.getMessage` call.
   For example, if the message key is **Message key text**, enter `${gs.getMessage("Message key text")}` in the form annotation.

### Administer form annotation types

You can define the form annotation types to control their appearance.

Role required: admin

1. Navigate to **System UI > Form Annotation Types**.
2. Set the **Active** field to **false** for any types you do not want to use.
3. Click **New** to add a type.

### Disable annotations

You can disable form annotations.

Role required: admin

1. Enter `sys_properties.list` in the navigation filter.
2. Locate the `glide.ui.form_annotations` property in the System Properties list.
3. Set the **Value** field to **false**.
   By default, this property is set to true to display annotations.

### Modify form focus

By default, forms set focus on the first writable field on the form. An administrator can change that functionality to focus on other fields in the form.

#### Modify form focus using a client script

Modify the form focus with `onLoad()` client script on the form.

```javascript
function onLoad ( ) {
    setTimeout ( "var refocus =
        g_form.getElement('table_name.field_name');refocus.focus();", 0 ) ; }
```

Based on the nature of the field you are targeting, replace `table_name.field_name` with the appropriate value:

- For non-reference fields, enter the name of the field to focus on (for example, `incident.caller_id`).
- For reference fields, use the format `sys_display.table_name.field_name` (for example, `sys_display.incident.caller_id`).
Disable form focus

Disable form focus by setting the property glide.ui.focus_first_element to false. To do so, navigate to System Properties > UI Properties and deselect the On form load, focus on first writable element check box.

Change the preference to submit a form with the enter key

By default, pressing the Enter key in a simple, one-line, choice list, or a Boolean field submits the form.

Role required: admin

A system preference controls this behavior, and it can be deactivated.

1. From the left navigation pane, select User Administration > User Preferences.
2. Select the enter_submits_form preference.
3. Set the value to false.
4. Click Update.

The change does not take effect until user preferences are reloaded either at login or when a session is created.

Multiple form splits

Form splits enable you to organize fields on a form into columns. When you use multiple form splits, you are grouping sets of fields together, one columnar group below another columnar group. A system property controls the use of multiple form splits.
Figure 226: Configuration of two form splits
When you organize fields in this manner and the user is viewing the form on a small mobile device, the fields within the first split are listed before the fields in the second split. In this example, the Asset tag, State, Serial number, and Substate field are listed before any of the fields below them. You can also create elements that span the form at the top of the form.

The property glide.ui.form_multiple_splits must be enabled to use this capability. If this property is missing or false, then things work as they always have. If this property is set to true, the following options appear in the Available column of the Configure Form slushbucket:

- |- begin_split -|
- |- split -|
- |- end_split -|

When you move any of these options to the Selected column, the names remain in the Available column, so you can select them multiple times.

**Note:** Additional split sections beyond the default split section must contain all three items: the begin, the split, and the end.

Add a property to enable multiple form splits

You can create the property to enable multiple form splits if it does not exist in your instance.

**Role required:** admin

1. Enter sys_properties.list in the Navigation filter.
The entire list of properties in the System Properties [sys_properties] table appears.

2. To determine whether the property exists, search for *multiple in the Name column. If the glide.ui.form_multiple_splits appears, you do not need to proceed with the following steps.

3. If the property does not exist, click New.

4. Enter the following information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.ui.form_multiple_splits</td>
</tr>
<tr>
<td>Description</td>
<td>Enable multiple form splits</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

5. Click the form context menu icon and select Save. The Categories related list appears.

6. Click Edit and move UI to the Categories List.

7. Click Save. The System Property form reopens and the new property appears in the UI Properties page.

Form section administration

Users with the form_admin role can divide forms into sections to organize the fields into logical groupings. Users can expand or collapse form sections to show or hide the fields they need. If you have tabs enabled, each form section appears on a separate tab. The default Change Request form is an example of a form with multiple sections as shown in the following image.
Figure 228: Change Request form sections

Create a form section

You can create a form section.
Role required: form_admin

1. Right-click the form header and select **Configure > Form Layout**.
2. In the **Form view and section** area below the slushbucket, click **New** in the **Section** list.
3. In the **Section caption** field, give the new section a title, then click **OK**.
4. Select a form section name and click the up or down arrow to change the section order.

   **Note:** The caption for the first section on the form becomes the form title.

5. Add fields to the new section using the slushbucket.
6. Click **Save** when you are finished.
   The new section appears on the form with the fields you selected. If you do not add any fields to a section, the section stays empty.

**Delete a form section**

You can delete a form section.

Role required: form_admin

1. Navigate to **System UI > Form Sections**.
2. Filter for the table that contains the section you want to delete. For example, the Incident table.
   The name of the form section to be deleted is displayed in the **Caption** field.
3. Select the check box beside the form section to delete.
4. From the **Actions on selected rows** menu, select **Delete**.
5. Click **OK** to confirm the deletion.

**Show or hide a form section**

When you show or hide a form section, your selection is saved as a user preference.

1. Open a form with multiple sections.
2. Perform the action appropriate to your version of the UI.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI16 or UI15</td>
<td>Click the expand (&gt;) or collapse () arrow on the section header. The expand and collapse arrows are visible only when tabbed forms are disabled.</td>
</tr>
<tr>
<td>UI11</td>
<td>Click the expand (+) or collapse (-) icon on the section header.</td>
</tr>
</tbody>
</table>

The next time you access a record that uses the same form, the same section is shown or hidden.

**Templates**

Templates simplify the process of submitting new records by populating fields automatically.

To use a template, populate the most-used fields for a specific table, save it as a template, and then make the template accessible to your users. Users can apply the template to new records on that table, or define scripts that apply the template.

Create templates for the forms that you use frequently, such as incident, problem, and change. There is no limit to the number of templates that you can create or access. Having many templates for each form makes the templates more complex to manage.
You can apply ACL rules to template creation. The user creating the template must satisfy the `save_as_template` ACL for any field modified by the template.

**Automatically applied templates**

You can make a template that automatically applies to new, user-created records on a specific table. To create this kind of template, set the template name to match the name of the table to which the template applies.

For example, imagine you want to apply a template automatically when a user creates a record on the Windows Server [cmdb_ci_win_server] table. Set the Name field to `cmdb_ci_win_server` and the Table field to Windows Server [cmdb_ci_win_server] on the Template form.

Automatic templates are always global. They do not honor the User and Group fields on the Template form. Automatic templates do not apply to records created by the system, such as those records generated by business rules, UI actions, or workflows.

**Toggle the template bar**

The template bar appears at the bottom of forms in UI16 and UI15. It provides shortcuts to apply, edit, and create templates.

Role required: none

You can toggle the template bar, which hides or shows it for all forms. The template bar is shown by default.

---

**Note:** The template bar is not available in UI11.

1. Navigate to a form.
2. Perform the appropriate action for your version of the UI.

| UI16 | 1. Click the more options icon (≡≡≡) in the form header.  
2. Select **Toggle Template Bar**. |
|------|----------------------------------------------------------------------------------------------------------------------------------|
| UI15 | 1. Right-click the form header.  
2. Select **Toggle Template Bar**. |

The template bar is hidden or shown.

Create a template using the Template form

You can create a template record for any table.

Role required: admin

1. Navigate to **System Definition > Templates**.
2. Click **New**.
3. Complete the form, as appropriate.

**Table 177: Template form fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Display name of this template.</td>
</tr>
</tbody>
</table>
| Table | Table this template applies to. Select **Global** to make the template available for use with all tables.  
Note: The list shows only tables and database views that are in the same scope as the template. |
| Active| Option for making the template available for use. A template must be active to be used. |
| User | User who can configure and apply the template. If a user is defined, no other users can see the template unless the **Global** check box is selected. |
| Group | A group whose members can configure and apply the template. If a group is defined, no other groups can see the template unless the **Global** check box is selected. |
| Global | Option for allowing any user who can access templates to view and apply this template. |
Field | Description
---|---
Short description | Description of the template.  
**Note:** Text added in this field does not add the text to the Short description field of forms that use this template.

Template | The content that automatically populates records based on this template. Select a field from the specified table in the left column, then enter the data to automatically populate in the right column.  
**Note:** Even though you can select dot-walked fields in the template, they do not apply to fields that are on the form.

4. Click **Submit**.

See **Application scope**.

Create templates for related task records

Administrators can create a template for a Task table record that also creates one or more related records in the child Task table.

Role required: admin

Administrators need to understand the parent-child relationships between Task tables. For example, the Change Task table is a child of the Change table and the Incident Task table is a child of the Incident table.

1. From the parent Task table template, **configure the form layout** to add these fields.
   - **Next Related Child Template**
   - **Next Related Template**
   - **Link element**

2. **Create a template** for the parent Task table.  
   For example, create this template for the Change table.

   **Table 178: Sample Change template**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Add server to network</td>
</tr>
<tr>
<td>Table</td>
<td>Change [change_request]</td>
</tr>
<tr>
<td>Short description</td>
<td>Set up a server on the network</td>
</tr>
</tbody>
</table>
3. **Create a template** for the first related task.
   For example, create this template for the Change Task table.

   **Table 179: First sample Change Task template**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Order server</td>
</tr>
<tr>
<td>Table</td>
<td>Change Task [change_task]</td>
</tr>
<tr>
<td>Short description</td>
<td>Order server hardware</td>
</tr>
<tr>
<td>Template</td>
<td>• [Short description][Order server hardware]</td>
</tr>
<tr>
<td></td>
<td>• [Assignment group][Hardware]</td>
</tr>
<tr>
<td>Link element</td>
<td>Change request</td>
</tr>
</tbody>
</table>

4. **Create a template** for each additional related task.
   For example, create one additional template for the Change Task table.

   **Table 180: Second sample Change Task template**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Install server on network</td>
</tr>
<tr>
<td>Table</td>
<td>Change Task [change_task]</td>
</tr>
<tr>
<td>Short description</td>
<td>Install server on network</td>
</tr>
<tr>
<td>Template</td>
<td>• [Short description][Install server on network]</td>
</tr>
<tr>
<td></td>
<td>• [Assignment group][Hardware]</td>
</tr>
<tr>
<td>Link element</td>
<td>Change request</td>
</tr>
</tbody>
</table>

5. From the parent Task table template, set **Next Related Child Template** to the first related child task.
   For example, add this value to the **Add server to network** template.

   **Table 181: First sample related template**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next Related Child Template</td>
<td>Order server</td>
</tr>
</tbody>
</table>
6. For each child related task, set **Next Related Template** to the next related task template. For example, add this value to the **Order server** template.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Next Related Template</strong></td>
<td>Install server on network</td>
</tr>
</tbody>
</table>

**Table 182: Second sample related template**

*Note:* The last related task template will not have a value for **Next Related Template**. For example, the **Install server on network** template will not have a value in **Next Related Template**.

Create a module for the parent Task table template. For example, create a module for **Add server to network**.

**Save a form as a template in UI11**

You can save a populated form as a template.

Role required: none

1. Navigate to a form, such as **Incident > Create New**.
2. Complete the form as it should appear when a user applies the template.
3. Right-click the form header and select **Save**.
4. Right-click the form header and select **Templates > Save as Template** to save only field values in the template, or **Templates > Save All as Template** to include related list relationships in the template.

*Note:* Read-only fields are not added to the list of fields on the template. If a field is made read only via ACL, UI policy/client script, or the dictionary, it does not populate in the template form creator.
After saving the template, the Template form appears.
5. Make any additional changes as needed using the Template form.
6. Click **Update** to save any changes to the template.

You can delete the original record, the incident in this example, after creating the template.

By default, the system displays 15 templates in the Apply Template context menu. Type `sys_properties.list` into the Application Navigator, then search for the `glide.template.max_context` system property to change it.

**Save a form as a template in UI16 or UI15**

You can save a populated form as a template.

Toggle the template bar so it is visible on forms.

**Role required:** none

1. Navigate to a form.
2. Complete the form as it should appear when a user applies the template.
3. In the template bar, click the plus icon (+).
4. Enter a descriptive name to make it easy for a user to select the correct template.
5. Make any additional changes as needed.
6. Click **Save**.
Schedule a template

You can define a scheduled job to create new records based on a template.

Role required: admin

For example, you can regularly create a populated task record to perform a weekly backup.

1. Navigate to System Definition > Templates.
2. Select a template record.
3. Click Schedule.
   The Scheduled Entity Generation form appears.
4. In the Run choice list, select how frequently to create a record.
5. Complete the schedule information.
6. Click Submit.

Create a module for a template

You can create a module to open a form with pre-populated template data.

Role required: admin

Child templates are only applied if the parent template is applied from a module. Child templates are not applied by applying a template to a new form.

1. Perform the appropriate action for your version of the UI:

<table>
<thead>
<tr>
<th>UI16</th>
<th>UI15 or UI11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Navigate to System Definition &gt; Application Menus.</td>
<td>Right-click an application menu, such as Self-Service, and select Edit Application Menu.</td>
</tr>
<tr>
<td>2. Open the application menu to which you want to add the module.</td>
<td></td>
</tr>
</tbody>
</table>

2. In the Modules related list, click New.
3. Fill in the Title and Order fields as appropriate.
4. In the Link Type form section, select the Table.
5. For the Link Type field, select New Record.
6. In the Arguments field, enter <table>.do?sys_id=-1&sysparm_template=<templatename>
7. Click Submit.
8. Refresh the application navigator to view the new module.

Scripted templates

You can apply an active template to a record using JavaScript.

**Apply a template to current**

To apply a template, use the applyTemplate method.

```javascript
current.applyTemplate("<templatename>");`
```
Apply a template to a GlideRecord

To apply the template to a record other than current, change current to a GlideRecord variable. When using a GlideRecord variable, you may need to initialize it after declaring the variable.

```javascript
var rec1 = new GlideRecord("incident");
rec1.initialize();
rec1.applyTemplate("my_incident_template");
```

Apply a template from a UI action

The following script demonstrates a possible customization to the Create Change UI action on the Problem form. After you add this script to the UI action, a user can select the UI action to create a change record with information from both the problem record and the change template.

```javascript
var change = new GlideRecord("change_request");
change.initialize();
change.short_description = current.short_description;
change.description = current.u_details;
change.cmdb_ci = current.u_service;
change.priority = current.priority;
change.requested_by = current.caller_id;
change.assignment_group.setDisplayValue('Change & Release');
change.u_status = 'New';
change.parent = current.number;
change.applyTemplate("standard_rfc");
current.rfc = change.insert();
current.comments = 'Change ' + change.number + ' created.';

var mySysID = current.update();

gs.addInfoMessage("Change " + change.number + " created");
action.setRedirectURL(change);
action.setReturnURL(current);
```

Script a template with child templates

When using applyTemplate with a template that has one or more child templates, the system creates the parent record before applying the child templates. This behavior ensures that any references or dot-walked fields from the child record to the parent have a valid target.

For example, if a template for the Change Request table has a child template for the Change Task table, applying the Change Request template inserts a Change Request record into the database. It assigns this record as the Change request for the Change Task record, then applies the child template to the Change Task record.

UI actions

UI actions add buttons, links, and context menu items on forms and lists, making the UI more interactive, customizable, and specific to user activities. UI actions can contain scripts that define custom functionality. Administrators and users with the ui_action_admin role can define UI actions.
UI action controls

You can create a UI action to provide any or all of these controls.

- A button on a form
- A context menu item on a form that appears when you open the form context menu or right-click the form header
- A related link in a form
- A button in the banner on top of a list
- A button at the bottom of a list
- A context menu item on a list that appears when you open the list context menu or right-click the list header
- A menu item for the action choice list at the bottom of a list
- A related link at the bottom of a list

**Note:** To hide or restrict the New or Edit UI action on a list or related list, use list control.
UI action visibility

You can use conditions to control UI action visibility by role. You can use the **UI Action Visibility** related list to restrict a UI action by view.

**Control UI action visibility by role**

Use this syntax to check for a role:

```java
gs.hasRole('role_name')
```

**Note:** This role is deprecated, starting with the Geneva release. See *Mandatory roles* on page 2527.

**Example: Require the admin role**

```java
gs.hasRole('admin')
```

**Example: Require the itil_incident or itil_problem role**

```java
gs.hasRole('itil_incident') ||
gs.hasRole('itil_problem')
```

**Example: Require the itil_incident and itil_problem role**

```java
gs.hasRole('itil_incident') && gs.hasRole('itil_problem')
```

**Control UI action visibility with views**

You can use the **UI Action Visibility** related list to restrict a UI action by view. A UI action is available for a given view according to the following rules.

1. If there are no visibility rules, the action appears on all views.
2. Any exclude rule on a given view means that the action does not appear on that view.
3. If there is at least one include rule, then the action appears only on views that are specifically included.

**Example: Show an action everywhere except the Cost Management view**

Add an exclude rule for the Cost Management view.
Example: Show an action on the Routine and ITIL views, but nowhere else

Add include rules for the Routine and ITIL views.
Edit a UI action

You can edit an existing UI action or create a new one.

Role required: ui_action_admin or admin

You can add UI actions to tables and database views that are in the same scope as the UI action and tables that allow UI actions from another application scope to run on them.

1. Navigate to System Definition > UI Actions.
2. Click New or open an existing record.
3. Define the UI action by completing the fields.
   You may need to configure the form to see all the fields.

Table 183: Editing UI actions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Defines the text that appears on the button, link, or context menu item.</td>
</tr>
<tr>
<td>Table</td>
<td>Defines the table on which the UI action is available. By default, the UI action also appears on tables that extend the selected table (for example, Task actions appear on the Incident table). Select Global to make the action available on all tables.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Order</td>
<td>Defines the order in which the UI action appears. The order applies to buttons from left to right and to menu actions from top to bottom.</td>
</tr>
<tr>
<td>Action name</td>
<td>Defines a name to use when referencing the UI action in scripts.</td>
</tr>
<tr>
<td>Active</td>
<td>Enables the UI action when selected. To disable a UI action, clear the check box.</td>
</tr>
<tr>
<td>Show insert</td>
<td>Shows a button on new records that have not been inserted.</td>
</tr>
<tr>
<td>Show update</td>
<td>Shows a button on existing records.</td>
</tr>
<tr>
<td>Client</td>
<td>The UI action executes its script in the user's browser, not on the server. When enabled, the Onclick field appears above the Condition field.</td>
</tr>
<tr>
<td>Form button</td>
<td>Puts a button on a form.</td>
</tr>
<tr>
<td>Form context menu</td>
<td>Puts a context menu item on a form (right-click the form header).</td>
</tr>
<tr>
<td>Form link</td>
<td>Puts a link in the Related Links section of a form.</td>
</tr>
<tr>
<td>List banner button</td>
<td>Puts a button in the banner of a list.</td>
</tr>
<tr>
<td>List bottom button</td>
<td>Puts a button at the bottom of a list.</td>
</tr>
<tr>
<td>List context menu</td>
<td>Puts a context menu item on list rows (right-click a row in a list).</td>
</tr>
<tr>
<td>List choice</td>
<td>Puts an item in the action choice list at the bottom of a list.</td>
</tr>
<tr>
<td>List link</td>
<td>Puts a link in the Related Links section at the bottom of a list.</td>
</tr>
<tr>
<td>Comments</td>
<td>Provides descriptive content regarding this UI action.</td>
</tr>
<tr>
<td>Hint</td>
<td>Defines the text that appears when a user points to the UI action control.</td>
</tr>
<tr>
<td>Onclick</td>
<td>The name of the JavaScript function to run when the UI action is executed. The function is defined in the Script field.</td>
</tr>
<tr>
<td>Condition</td>
<td>Defines the conditions that restrict when a UI action appears.</td>
</tr>
<tr>
<td>Script</td>
<td>Defines the script to run when the UI action is executed.</td>
</tr>
<tr>
<td>Requires role</td>
<td>Restricts the roles that have access to this UI control.</td>
</tr>
</tbody>
</table>

Related lists on the form view:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Action Visibility</td>
<td>Specifies other views of the form that the UI action applies to.</td>
</tr>
<tr>
<td>Versions</td>
<td>Shows all versions of the UI action. Use this list to compare versions or to revert to a previous version.</td>
</tr>
</tbody>
</table>

4. Click **Submit** or **Update**.

**Note:** If the UI action is enabled to run on the client side, wrap it in a function. Otherwise, the contents of the Script field run when the page loads.

Override or remove a UI action for an extended table

This example describes how to override or remove a UI action for a table that is extended from another table.

When a UI action is defined for the Task table, it applies to all tasks, including incidents, changes, problems, and any other tables that extend the Task table. Similarly, a global UI action applies to every table. However, you can override a UI action for a specific table. This example demonstrates how to override a UI action on the Task table for only the Incident table.

1. Complete the following steps to override a UI action on the Task [task] table for just the Incident [incident] table.
   a) Create a UI action on the Incident table with the same **Action name**.
      If the **Action name** is not defined, use the same **Name**.
   b) Enter a script that is specific to the Incident table.

2. Complete the following steps to remove a UI action on the Task [task] table for the Incident [incident] table.
   a) Navigate to the UI action definition for the Task table.
   b) Add the condition `current.getRecordClassName() !== 'incident'`.

UI action restrictions

When needed, you can restrict when a UI action is available.

For example, you might want to show **Close Incident** only to group managers or only for incidents that are already in the **Resolved** state. Alternatively, you might want to display an action only on certain views of a form.

**Note:** In UI11 only, you can control whether the UI action appears at the bottom of the form in addition to the top. **Create the property** `propertyglide.ui.buttons_bottom` and set it to **true**.

Control visibility with conditions

The most direct way to control the availability of an action is to define the **Condition** field. The condition must evaluate to **true** for the action to appear. If you leave the field empty, the condition defaults to **true**.

For example, the following condition is configured for the **Close Incident** button.
Figure 233: UI action condition

For this action to appear on a form, these conditions must evaluate to true:

- current.incident_state must equal 6: The incident must already be in a Resolved state.
- gs.hasRole("itil_admin"): The current user must have the itil_admin role.

OR this condition must evaluate to true:

- gs.GetUserID() == current.caller_id: The current user is the user who requested the change.

**Note:**

- The current object is not available for conditions on a list context menu (the List context menu check box is selected). Any use of current on these actions is ignored.
- You can reference the parent record for the UI action conditions on a related list button. For example, to disable the **New** and **Edit** buttons on the Affected CIs related list for closed changes, copy the global m2m UI actions to the task_ci table and add a condition of `parent.active`.

---

**URL redirection using a UI action**

As part of a UI action, you can redirect a user to a URL.

For example, you can add links to a form or open a new record after it is created from a UI action.

To redirect a user to a URL from a UI action, use this syntax to define the redirect link:

```java
action.setRedirectURL ( 'http://www.mysite.com/mypage.htm' ) ;
```

To direct a user to a record, use this syntax, where `new_record` is the variable name for the **GlideRecord**:

```java
action.setRedirectURL ( new_record ) ;
```

**Tabbed forms**

Tabbed forms offer a useful way to make forms and related lists take up less space by reducing the scrolling that must be done to navigate the form.
Figure 234: Tabbed forms
Form sections and related lists are tabbed separately, each with their own tab line. A user always sees the first form section. All sections after that can be tabbed. Tabs are enabled by default for new instances.

Display tabbed forms

Users can enable tabbed forms in their user preferences.

Tabbed forms are enabled by default for new instances. A system user preference with the name `tabbed.forms` specifies whether the tabbed UI is used by default for all users. Users can change this preference as described in this procedure.

1. Click the gear icon in the banner frame.
2. Perform the appropriate action for your version of the UI:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI16</td>
<td>Select the Forms tab and enable Tabbed forms.</td>
</tr>
<tr>
<td>UI15</td>
<td>Enable Tabbed forms.</td>
</tr>
<tr>
<td>UI11</td>
<td>Click the tabs toggle icon, which appears on any form with more than two sections and on any form with more than one related list.</td>
</tr>
</tbody>
</table>

In UI16 and UI15, an option is enabled when the switch appears green and is toggled to the right.

Require fields on a form

When Required Form Fields is activated, an administrator can designate specific fields on a form as required (protected) fields. Users cannot hide required fields when they personalize the form.

If present on a form, only an administrator can remove required fields. When you configure a form, required fields appear in a gray color and have a tooltip indicating they are required. Required fields are defined in the Required Form Fields [sys_ui_element_required] table.

The expected (although not the only) use case for this feature is as part of a delegated administration scheme. For example, you can grant branch offices the rights to modify forms by granting the personalize_form role, but not allow them to remove certain fields which are critical to overall business processing.

Activate required form fields

You can activate the Required Form Fields plugin (com.snc.required_form_fields) if you have the admin role. This plugin activates related plugins if they are not already active.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.

3. Optional: If available, select the Load demo data check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click Activate.
Define required fields

You can specify which form fields are required in forms. This action prevents users with the personalize_form role from removing the field by configuring the form layout.

Activate the Required Form Fields plugin before you perform this task.

Role required: admin

1. Navigate to sys_properties.list.
2. Locate the property named glide.ui.form.enforce_required_fields and make sure it is set to true.
3. Navigate to System UI > Required Form Fields.
4. Click New.
5. Select the table and field, and then select the Required check box.
   The following example sets the Short description field on the Incident form to be required.

![Required Form Field](image)

If you decide later that you do not want to make the field required, clear the Required check box. This action is preferable to deleting the record.

6. Click Submit.

If the table specified has extension tables, then the Required Form Field record applies to forms of all extended tables. For example, if an administrator specifies that the Short description field is required for the Task table, then this configuration applies to the Incident form, Change Request form, Problem form, and so on.

An extended table can override the base table Required Form Field rule. For example, if the Short description field is required for the Task table, but not required for the Incident table, it is required for all Task tables except Incident.

Display information from other records

Administrators can display fields that are associated with another field.

For example, on the Incident form, the name of the submitter appears in the Caller field. You can configure the Incident form to show the caller's city. The city is derived from the User table using the value in the Caller field to determine which record is correct.
When you configure a form, the fields that have derived fields associated with them appear in green with the + symbol next to them. Click the field, and then click the expand icon to show the other associated fields that can be added to the form. The following example shows the derived fields, based on the Caller, that can be added to an Incident form.

**Figure 235: Available table fields**

Choose the **Caller** field and click the expander icon.

**Figure 236: Caller fields**

Select the available fields associated with the **Caller** field.

### Control the label type for derived fields

A derived field is a field that is not in the primary table that is being displayed. A derived field is obtained by following a reference from the primary table to another table. You can configure the type of label that appears for derived fields.

**Role required:** admin

An example would be displaying the caller's email address when looking at an incident record. The caller's email address is not stored in the Incident table, but is obtained by following the caller reference from the incident to the User table.
You control the label type by setting a system property. For example, the field label for an incident caller's email address can be either of the following strings.

- **Caller Email**, which represents the complete label to uniquely identify this field as the email from the caller's user record. If you display the email address for the person the ticket is assigned to, its label would be **Assigned to Email**.
- **Email** which is the label for the target field. This label is not unique on the form if, for example, you are also displaying the email address of the person assigned to the incident. However, usually the placement of the field on the form makes it clear what the field represents.

1. Navigate to **System Properties > System**.
2. Locate the **Use short labels for all fields. For example, if a form contains the caller's email address, use the "Email" label rather than the full label of "Caller Email"** (glide.short.labels) property.
   - The default value is **true**, meaning that the short label is displayed in all forms.
3. If you want to display the complete label in forms, clear the check box.
4. Click **Save**.

**Attachment administration**

Administrators can configure options for how files are uploaded and attached to records.

**Note:** To learn how to upload attachments to records, see *Add attachments* on page 127.

**Attachment tables**

When you store an attachment to any table, a record is created in the Attachment [sys_attachment] table that contains attachment metadata, and the file data is stored in the Attachment Document [sys_attachment_doc] table, in 4k chunks. For example, if you attach a 12k file called *My_attachment.pdf*, then there is an Attachment entry with three related Attachment Document entries.

**Disable attachments on a table**

You can prevent attachments from being added to records on a specific table.

Role required: admin

1. Open a record in the table.
2. Right-click the form header and select **Configure > Dictionary**.
3. In the list of dictionary entries, select the first record in the list (the record with no **Column name** entry).
4. Add **no_attachment** to the **Attributes** field, separated by commas from any existing attributes.
   - See *Dictionary attributes* on page 1453 for more information.

**Disable the drag and drop feature**

You can disable the drag-and-drop attachment feature.

1. Navigate to **System Properties > UI Properties**.
2. Clear the check box for the **Allow attachment drag and drop in supported HTML5 browsers** property.
3. Click **Save**.
Limit attachment file size

You can specify the maximum attachment size.

2. Enter a value in the Maximum file attachment size in megabytes property. Leave the field empty to allow attachments up to a maximum of 1GB. By default, this field is blank.
3. Click Save.
   The maximum attachment size for email attachments is configured separately.

   Warning: Uploading large attachments might lead to issues with the user's active session on the instance.

Restrict who can attach files

You can restrict who can add attachments by using a role.

2. In the Attachment limits and behavior section, locate the List of roles (comma-separated) that can create attachments property (glide.attachment.role).
3. Enter one or more roles separated by commas.
   Only roles listed in this property are able to upload attachments to a record. If no roles are entered, then all roles can upload attachments.
4. Click Save.

Restrict file extensions

The glide.attachment.extensions property restricts the file extensions that users can upload as attachments.

Use this property to improve security by preventing users from uploading harmful files, such as viruses, as attachments. Additionally, this capability can prevent the use of incompatible filetypes. For example, Internet Explorer does not support icons in .png format. Note that this property does not restrict files based on the actual file type, but only based on the extension.

2. In the Attachment limits and behavior section, locate the List of file extensions (comma-separated) that can be attached to documents via the attachment dialog property.
3. Enter the file extensions and click Save.
   If no extensions are specified, then all extensions are allowed. However, if any extensions are specified, all unlisted extensions are restricted. Listed extensions should not include the dot (.) prefix or spaces after commas. For example, enter xls,xlsx,doc,docx.

Hide the attachment [view] link

Users can open an attachment by clicking either the file name or the [view] link. The [view] link opens the file from within the browser, which executes JavaScript code as part of the attachment.

1. Add the glide.ui.disable_attachment_view and glide.ui.attachment_popup properties.
For instructions on adding properties to the platform, see Add a property on page 1411.

2. For the glide.ui.disable_attachment_view property, set the Type to true/false and set the Value to true.

3. For the glide.ui.attachment_popup property, set the Type to true/false and set the Value to false.

4. To return to the default behavior (enable the link), set the glide.ui.attachment_popup property Value to true.

**Note:** Users can still view attachments by clicking the file name.

### Configure attachment icons

You can configure the icon that appears beside an attachment of a particular file type.

1. Determine the path of the image file. If desired, upload a new image.

2. Navigate to **System UI > Attachment Icon Rules**.

3. Open an existing rule or click **New** to create a new rule.

4. Enter the rule details.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select icon by</td>
<td>Select MIME Type or File Extension.</td>
</tr>
<tr>
<td>MIME type</td>
<td>If <strong>Select icon by</strong> is MIME Type, enter the MIME type and subtype to associate with the icon, separated by a slash (example: application/pdf).</td>
</tr>
<tr>
<td>File extension</td>
<td>If <strong>Select icon by</strong> is File Extension, enter the file extension to associate with the icon beginning with the period (example: .pdf).</td>
</tr>
<tr>
<td>Icon</td>
<td>Enter the path to the icon image file (example: images/icons/attach_pdf.gifx)</td>
</tr>
</tbody>
</table>

5. Click **Submit** or **Update**.

### Attachment events and logging

Uploading, viewing, and deleting file attachments triggers a single event that can be used for notifications or in scripts. Attachment events can be processed by a script action or notification.

Only one event is created when action is taken on a record with attachments, even when the record has multiple attachments. The following events are provided.

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment.read</td>
<td>An attachment has been read or downloaded.</td>
</tr>
</tbody>
</table>
### Event logging

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment.uploaded</td>
<td>An attachment has been uploaded. If multiple attachments are uploaded to a record at one time, only one event is created.</td>
</tr>
<tr>
<td>attachment.deleted</td>
<td>An attachment has been deleted. This event is also triggered when the record containing the attachment is deleted. If a record is deleted that contains multiple attachments, a separate event is triggered for each attachment in the deleted record.</td>
</tr>
<tr>
<td>attachment.renamed</td>
<td>An attachment has been renamed.</td>
</tr>
</tbody>
</table>

### Attachment logging

When an attachment is downloaded, the attachment.read event record is written, and you can do something with this event. For example, you can record when and by whom certain attachments are downloaded. For this functionality, `current` is a `sys_attachment` record, and the event record uses the following parameters:

- `parm1`: File name
- `parm2`: Table name

### Enable attachment indexing on a table

When you enable the attachment indexing for a table, text searches can return matches from the record and its file attachments.

Role required: admin

By default, attachment indexing is enabled for the Knowledge Base.

Enabling attachment indexing causes the platform to re-index the selected table, its parent table, and any children of the parent table.

**Warning:** For large tables, such as the Task table, re-indexing can take several hours and slows down the system until complete. Re-indexing is best performed during non-peak times.

1. Navigate to **System Definition > Dictionary** and select the record for the table.
2. Click the **Attributes** tab.
3. Click **New** and add the following values.

### Table 185: Attribute values

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Attachment index</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

**Note:** The attachment indexing attribute only applies to the tables on which you explicitly add it. It does not cascade to child tables. For example, indexing attachments on the Task table does not index attachments on the Incident table.
4. Click **Submit**.
5. Optional: To disable attachment indexing, remove the attribute.
6. Click **Update**.
   The indexing process begins. When it is complete, attachments can be searched on that table.

**Debug attachment indexing**

To change debugging options for attachment indexing, add these system properties.

- `glide.ts.index.attachment.debug`: when the value is set to **true**, enables log messages for exceptions that occur when indexing attachments (default is **false**).
  You can leave this property enabled during normal operations to capture stack trace information about any exceptions.
- `glide.ts.index.attachment.list_terms.debug`: when the value is set to **true**, logs all indexed terms when an attachment is indexed (default is **false**).
  [Recommended] For optimal performance, set this property to **false** during normal operations. Only enable this property when you are actively debugging an issue.

**Create a UI policy**

UI policies offer an alternative to client scripts for dynamically changing information on a form. Use UI policies to define custom process flows for tasks.

Role required: **ui_policy_admin**

All fields are evaluated by a UI policy condition even if they are not visible on the form (in UI16 and UI15 only). This function removes the requirement that a field must be on a form for it to be evaluated. In UI11, a UI policy condition only evaluates fields that are present on a form. Fields that are not present are not evaluated. For UI11, ensure that any field called by a UI policy exists on the form.

- **Note**: Policies carried over from versions prior to Fuji are evaluated differently. Fields that previously were not evaluated are evaluated.
- **UI policies are not supported on search screens.**

1. Navigate to **System UI > UI Policies**.
2. Click **New**.
   The UI Policy [Advanced view] form opens
3. Optional: To change the view, in **Related Links** click **Default view**.
4. Complete the form, as appropriate.
   You may need to configure the form to see all the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>The table for the form to be modified.</td>
</tr>
<tr>
<td>Active</td>
<td>The UI policy active status. Only active UI policies are applied.</td>
</tr>
<tr>
<td>Short description</td>
<td>Short summary of the UI policy.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Order [Advanced view]</td>
<td>The processing sequence, from lowest to highest number. If two policies conflict, the UI policy with the higher number is executed.</td>
</tr>
<tr>
<td>When to Apply</td>
<td></td>
</tr>
<tr>
<td>Conditions</td>
<td>The conditions which, if fulfilled, cause the UI policy to be applied. Conditions are built with the condition builder. To set conditions using a script, use a client script instead. Conditions are only rechecked if a user manually changes a field on a form. If the change is made by a UI action, context menu action, or through the list editor, it is not evaluated. In UI11, the field must be on the form to be checked by a UI policy. To test the value of a field but hide it from users, add the field to the form and use UI policies to hide it. This requirement also prevents dot-walked fields from being used as conditions.</td>
</tr>
<tr>
<td>Global [Advanced view]</td>
<td>Option for specifying whether the UI policy applies to all form views. If this check box is cleared, the UI policy is view-specific.</td>
</tr>
<tr>
<td>View [Advanced view]</td>
<td>Option for indicating which view or views the UI policy applies to. This option is visible only if Global is not selected.</td>
</tr>
<tr>
<td>Reverse if false [Advanced view]</td>
<td>Option for specifying that the UI policy action is reversed when its conditions evaluate to false. In other words, when the conditions are true, actions are taken and when they change back to false, the actions are reversed (undone).</td>
</tr>
<tr>
<td>On load [Advanced view]</td>
<td>Option for specifying that the UI policy behavior is performed OnLoad as well as when the form changes.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Inherit [Advanced view]                    | Option for specifying whether extended tables inherit this UI policy. When a child table has an inherited UI policy from its parent table, the UI policy on the child table always runs first. This event is true regardless of the Order of the UI policies. Consider the following example:  
  - A child table has a UI policy with Order value 500 that shows the Urgency field when its conditions are met.  
  - Its parent table has a UI policy with the same conditions that hides the Urgency field. The parent table UI policy has Order value 100.  
  - Although the parent table Order field has a lower value, the child UI policy runs first and then the parent UI policy runs. When the conditions are met, the Urgency field is hidden. |

**Script**

<table>
<thead>
<tr>
<th>Run scripts [Advanced view]</th>
<th>Option for specifying whether advanced behavior can be scripted for both true and false conditions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute if true [Advanced view]</td>
<td>A script that executes if the UI policy conditions are fulfilled. This field is available only if Run scripts is selected.</td>
</tr>
<tr>
<td>Execute if false [Advanced view]</td>
<td>A script that executes if the UI policy conditions are not fulfilled. This field is available only if Run scripts is selected.</td>
</tr>
</tbody>
</table>

**Other fields**

<table>
<thead>
<tr>
<th>Run scripts in UI type</th>
<th>The UI type for this UI policy: Desktop, Mobile, or Both.</th>
</tr>
</thead>
</table>

**Related List: UI Policy Actions**

<table>
<thead>
<tr>
<th>Table</th>
<th>Table the UI policy action executes on.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field name</td>
<td>Field on the selected table that the UI policy performs an action on if conditions are met.</td>
</tr>
</tbody>
</table>
| Mandatory                                  | Choice list for specifying how the UI policy affects the mandatory state of the field. Choices are:  
  - Leave alone  
  - True  
  - False |
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Visible         | Choice list for specifying how the UI policy affects the visible state of the field. Choices are:  
• Leave alone  
• True  
• False         |
| Read only       | Choice list for specifying how the UI policy affects the read-only state of the field. Choices are:  
• Leave alone  
• True  
• False         |

### Related Links

| Default view or Advanced view | Changes the form view to the default or advanced view. The fields change based on the view. |

5. Click **Submit**.

**Example: Create a UI policy**

Create a UI policy to implement controls in the Incident form when the state changes to **Resolved**.

**Role required: ui_policy_admin**

This example demonstrates how to implement the following controls.

- Make a **Close Notes** field mandatory.
- Hide the **Opened by** field.
- Make the **Priority**, **Severity**, and **Urgency** fields read-only.
- Run a client script that displays an alert message.

1. Navigate to **System UI > UI Policies**.
2. Click **New**.
3. Supply the following information.

### Table 187: New UI policy

<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Incident</td>
</tr>
<tr>
<td>Conditions</td>
<td>[Incident state] [is] [Resolved]</td>
</tr>
<tr>
<td>Reverse if false</td>
<td>Select this check box. If the incident state is not <strong>Resolved</strong>, the UI policy is reversed.</td>
</tr>
<tr>
<td>On load</td>
<td>Select this check box to perform the actions when the form is loaded or when the condition changes.</td>
</tr>
</tbody>
</table>
4. Right-click the form header and select **Save** from the context menu. The **UI Policy Actions** related list appears.

5. In the related list, click **New**.

6. Provide the following information.

   **Table 188: New UI policy actions**

<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field name</td>
<td>Close notes. This UI action makes the Close notes field mandatory.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>True</td>
</tr>
<tr>
<td>Visible</td>
<td>Leave alone</td>
</tr>
<tr>
<td>Read Only</td>
<td>Leave alone</td>
</tr>
</tbody>
</table>

7. Click **Submit**.

8. Repeat the process to create UI policy actions to hide the **Opened by** field, and to make the **Priority**, **Severity**, and **Urgency** fields read-only.

**UI policy options**

You can use different options in the UI Policy form to control when and how the UI policy is applied. These options include client scripts, OnLoad execution, and view-specific UI policies.

**Client scripts for UI policies**

Administrators can use the UI Policy form to create client scripts that run onChange when the UI policy conditions are met (**Execute if true**) or not met (**Execute if false**). To display these scripting fields in the UI Policy form, in the Script section, select the **Run scripts** check box.

For example, to display an alert to the user when the incident **State** field changes to **Resolved**, create the following script in the **Execute if true** field.

```javascript
function onCondition(){
    alert('You changed the "Incident state" to Resolved. Please enter your comments in the "Close notes" field.');
}
```

**On load check box**

You can check or clear the **On load** check box in a UI policy to control whether it runs every time a form is loaded when the conditions are satisfied. In this example, an administrator does not want an incident to enter the **Awaiting user info** state unless the user provides an explanation to the customer. The administrator creates a UI policy with the following settings.

- In the **When to Apply** section, adds the condition **[State] [is] [Awaiting user info]** and clears the **On load** check box. This condition means that the UI policy applies only when the state is changed to **Awaiting user info**.
- In the **UI Policy Actions** related list, creates a record that makes the **Additional comments** field mandatory when the condition is met.
If the administrator selected the **On load** check box, the UI policy would apply every time a user opens an incident in the **Awaiting user info** state.

**View-specific UI policies**

By default, the **Global** UI policy applies to all form views. However, a UI policy can be specific to a view. For example, you can define a UI policy for only the itil view of a form. Two fields on the UI Policy form control who can view a form.

- **Global**: If the **Global** check box is not selected, the **View** field is displayed.
- **View**: Select a view and the UI policy applies only to that view of the form.

**Convert a UI policy to a data policy**

To make a UI policy the default setting, convert the UI policy to a data policy.

Role required: `ui_policy_admin`

You can also apply a UI policy to import sets or to data imported by SOAP web services when you convert it to a data policy. Converting a UI policy to a data policy deactivates the UI policy. To retain the policy in the UI, ensure that the **Use as UI Policy on client** check box is selected on the data policy record.

For a UI policy to be eligible for conversion to a data policy, the following conditions must be met on the UI Policy form.

- The **Run scripts** check box must be cleared.
- The **Global** check box must be selected.
- None of the UI policy actions can have **Visible** set to **True** or set to **False**. **Visible** must be set to **Leave Alone**.

1. Navigate to **System UI > UI Policies**.
2. Open an existing UI policy.
3. Under **Related Links**, click **Convert this to Data Policy**. A new data policy record is created.
4. Edit the fields on the data policy record as necessary.

**Formatters**

A formatter is a form element used to display information that is not a field in the record. Add formatters to a form by configuring the form.

**Table 189: Examples of formatters in the base platform**

<table>
<thead>
<tr>
<th>Formatter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity formatter</td>
<td>Displays the list of activities, or history, on a task form.</td>
</tr>
<tr>
<td>Process flow formatter</td>
<td>Displays the different stages in a linear process flow across the top of a record.</td>
</tr>
<tr>
<td>Parent breadcrumbs formatter</td>
<td>Provides breadcrumbs to show the parent or parents of the current task.</td>
</tr>
<tr>
<td>Approval summarizer formatter</td>
<td>Displays dynamic summary information about the request being approved.</td>
</tr>
<tr>
<td>Formatter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CI relations formatter</td>
<td>Displays on the CI form a toolbar for viewing the relationships between the current CI and related CIs.</td>
</tr>
</tbody>
</table>

Creating a custom formatter involves the following three tasks.

1. Create a UI macro to define content for the formatter.
2. Create a formatter that refers to the UI macro.
3. Add the formatter to the form.

**Note:** Formatter elements cannot be exported to PDF. When exporting PDF data from a form, any formatter elements added to the form will not display in the PDF output.

Create a UI macro for a formatter

Create a UI macro to define the content that the formatter displays.

This functionality requires a knowledge of Jelly script.

Role required: ui_macro_admin

1. Navigate to System UI > UI Macros.
2. Click New.
3. Complete the form.
4. In the XML script field, enter Jelly script to define the content to be displayed by the formatter.

**Note:** The UI Macro for the formatter represents a row in the UI. It must begin and end with <TR></TR> tags.
This Jelly script is reproduced below in plain text that you can copy into the Macro form as a basis for your macro script.

```xml
<?xml version = "1.0" encoding = "utf-8" ?>
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:g2="null">
<tr>
    <td colspan="2">
        Here is some descriptive text for the Priority field.
    </td>
</j:jelly>
```

5. Click **Submit**.

Create a formatter and add it to the form

Create the formatter after creating the UI macro that defines the content.

Role required: admin
With the UI macro in place, create the formatter that references it, and add it to the form.

1. Navigate to **System UI > Formatters**.
2. Click **New**.
3. In the **Name** field, enter a descriptive name for the formatter.
4. In the **Formatter** field, enter the name of the UI macro with `.xml` appended to it.
5. In the **Table** field, select the table in which the formatter is used.
6. In the **Type** field, select **Formatter**.
7. Click **Submit**.
8. To add the formatter to a form, **configure the form**.
   The formatter appears on the form according to your selections.

### Override a formatter with macros

The platform uses formatters to handle complex rendering of specific form elements.

**Role required:** `ui_macro_admin`

Examples of form elements rendered by formatters in the base platform are:

- **Activity formatter**: Displays the list of activities, or history, on a task form.
- **Process flow formatter**: Displays the different stages in a linear process flow across the top of a record.
- **Task parent breadcrumbs formatter**: Provides breadcrumbs to show the parent or parents of the current task.
- **Approval summarizer formatter**: Displays dynamic summary information about the element being approved.

Formatters provided in the base system can be overridden by a UI macro.

1. Navigate to **System UI > UI Macros**.
2. Click **New**.
3. In the **Name** field, enter the same name as the formatter you want to override, but omit the `.xml` extension.
4. Complete the remaining fields on the form.
5. Click **Submit**.

This example shows the form that defines the existing approval summarizer formatter:
Here is the form for the UI macro that overrides the approval summarizer formatter:

Approval summarizer formatter

The approval summarizer formatter creates the summary at the bottom of an approval form.

The approval summarizer displays different information depending on what is being approved, such as a change request or a service catalog request. Following are two examples.
Summary of Item being approved

Change Request

Number: CHG0000001
Requested by: David Loo

Affected CI: Sales Force Automation
Type: Normal

Planned start date: 2016-07-27 16:00:00
Risk: High

Planned end date: 2016-07-27 18:00:00
Impact: 3 - Low

Short description: Rollback Oracle Version

Description: Performance of the Siebel SFA software has been severely degraded since the upgrade performed this weekend.

We moved to an unsupported Oracle DB version. Need to rollback the Oracle Instance to a supported version.
Figure 238: Catalog request approval summary

The **Reject** button allows the approver to reject one or more requested items in a multi-item request, before approving the overall request. If a requested item is rejected, the workflow for that item never starts. The approver can then choose to **Approve** the item.

**Note:** When the overall request is approved, you must ensure this **Approve** button is hidden. If this button is used after request approval, the requested item workflow is canceled, leaving the stage in an inconsistent state.

Similarly, the **Approve** button on requested items should only appear before the overall request is approved or rejected.

Activity formatter

The activity formatter provides an easy way to track items not saved with a field in the record, for example, journal fields like comments and work notes.

The activity formatter is enabled by default on the Task [task] table and other tables that extend the Task table, such as the Incident [incident] table. It is also enabled on the Approvals [sysapproval_approver] table.

You can filter the content that appears on the activity formatter and participate in the record feeds on the record. You can create an activity formatter for any form and configure properties that control what fields appear in the formatter.

In UI16, the activity formatter shows updates in real time so you can see the latest information without refreshing the form. User presence enables you to see when other users are entering comments.
Figure 239: UI16 activity formatter
Add the activity formatter to a form

You can add an activity formatter to any form.

Role required: personalize_form

1. Verify that the table associated with the form is audited.
2. Configure the form layout to add **Activities (filtered)**.

Enable the Live Feed-Activity toggle

The Live Feed-Activity toggle allows users to switch between the activity feed and the document feed for a record.

Role required: admin

Use live feed to interact with other users on a record by posting messages and adding attachments to the feed. Use the activity formatter to see an overall summary of activity for the record.
Figure 241: Live Feed/Activity toggle

- The glide.ui.show_live_feed_activity property is enabled. Navigate to Collaborate > Feed Administration > Properties and enable the Toggle the display of the live feed tab in the activity formatter option.

1. Verify that the live feed and record feed plugins are active.
2. View the form to ensure the activity formatter is visible. If not, configure the form to add it.
3. Set the live_feed dictionary attribute to true on the form. This adds live feed to the activity formatter.
4. Complete the following steps to enable the system property.
   a) Navigate to Collaborate > Feed Administration > Properties.
   b) Enable the Toggle the display of the live feed tab in the activity formatter option (glide.ui.show_live_feed_activity property).

Customize activities
You can customize which fields appear in the activity formatter. You can add or remove fields from the list of activities that users can select when they open the activity filter.

Role required: personalize_form
1. Scroll to the activity stream and perform the appropriate action for your version of the UI:

| UI16 | 1. Click the activity filter icon ( ).
|      | 2. At the bottom of the list, click **Configure available fields**. |
| UI15 or UI11 | Right-click the **Activity** header and select **Configure Activities**. |

2. In the slushbucket, select the desired fields in the **Available** column and move them to the **Selected** column.

   **Note:** The activities appear in alphabetical order, regardless of the order in the **Selected** column.

3. Click **Save** to add them to the activity formatter and the filter.

Administrators can also modify the system property **Incident activity formatter fields** (glide.ui.incident_activity.fields). Access this property through **System Properties > UI Properties**.
The system automatically synchronizes the values in the system property and the selections made through Configure Activities.

Create an activity formatter
You can create an activity formatter for any audited table.

Role required: admin

1. Navigate to System UI > Formatters.
2. Click New.
3. Enter a name for the formatter, such as Activities (task).
4. Select a Table.
5. Enter activity.xml in the Formatter field.
6. Leave the Type as Formatter.
7. Click Submit.

**Note:** You can create more than one activity formatter for a table, however, the system does not allow you to add more than one activity formatter to a form.

Add the new activity formatter to forms as needed.

Configure the email property for activity formatter
The system property glide.ui.activity.email_roles enables you to control which roles can see emails in the activity formatter.

Role required: admin

In the activity formatter, users see activity only for fields they have permission to read. For example, self-service users might see the activity formatter on the self-service view of the Incident form, but they do not see work notes, unless security rules have been customized to allow this.

If the Sent/Received Emails field is included in the activities list, all users see all emails. No determination is made whether an end user, for example, should see an email containing work notes. Configure this property to restrict this capability to specified roles.

**Note:** Email does not appear as an activity until it is sent. If email properties are not configured for outbound delivery, the message can be found by navigating to System Mailboxes > Outbox.

1. Navigate to System Properties > UI Properties.
2. Locate the property labeled List of roles (comma separated) that can view emails in the Activity Formatter when "Sent/Received Emails" are included.
3. Add roles to the property, separated by commas.
   These are the only roles that can see email in the activity formatter. All other roles are prevented from seeing email. If no roles are listed, all users can see email. The itil role is on the list by default.
4. Click Save.

Process flow formatter
The process flow formatter provides a graphical summary of the stages in a process. The formatter is typically shown at the top of forms that are part of a process.

Each record on the Flow Formatter [sys_process_flow] table represents a process stage and can have a different condition applied to it. When specified conditions are fulfilled, the formatter highlights the current stage and all previous stages.
These examples show a workflow in the UI15 and UI11 interface.

**Figure 243: UI15 process flow for an incident**

As soon as any formatter stages are defined for a table, they appear on the form associated with that table in the order specified, assuming the formatter has been added to the form.

**Add a process flow formatter**

To add a process flow formatter, complete these tasks.

1. Create a process flow formatter.
2. Attach the process flow formatter to the form.

**Create a process flow formatter**

You can create a process flow formatter stage.

Role required: admin

1. Navigate to **System UI > Process Flow**.
2. Click **New**.
3. Complete the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select a table for this process flow formatter.</td>
</tr>
<tr>
<td>Name</td>
<td>Enter a name to identify the formatter.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Label</td>
<td>Enter the name to be displayed in the Personalize Form Layout slushbucket.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to ensure the formatter stage is active. When the check box is cleared, the formatter stage does not appear in the flow display.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to indicate where in the process flow the formatter will be displayed. Formatters are arranged with the lowest number on the left and the highest number on the right.</td>
</tr>
<tr>
<td>Condition</td>
<td>Use the condition builder to set the conditions under which the formatter is highlighted as current. Any field available in the condition builder, such as SLA or Impact, can be used to trigger a process flow stage.</td>
</tr>
<tr>
<td>Description</td>
<td>Describe the process flow formatter stage in this HTML field.</td>
</tr>
</tbody>
</table>

4. Repeat as necessary for each stage.

Attach the process flow formatter to the form

After creating the formatter, attach it to the form.

Role required: personalize_form

Configure the appropriate form to attach the new formatter.
Activate the process flow formatter

You can activate the Process Flow Formatter plugin.

Role required: admin

1. Navigate to System Definition > Plugins.

2. Right-click the plugin name on the list and select Activate/Upgrade.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.

3. Optional: If available, select the Load demo data check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click Activate.

Add the parent breadcrumbs formatter to a form

The parent breadcrumbs formatter on the Task table provides breadcrumbs that show the parent or parents of the current task.

Role required: personalize_form

This formatter can be used also on any table that extends Task.

To add the parent breadcrumbs formatter to a form, configure the form and add Parent Breadcrumbs to the desired location. The breadcrumbs show only six levels of parents. If more levels exist, the breadcrumbs display an ellipsis ("...").
The Parent reference field also has to contain a value for the breadcrumbs to appear. You may need to configure the form to contain the Parent field as well.

**Figure 245: Breadcrumbs**

**Customize the parent breadcrumbs formatter**

You can customize the parent breadcrumbs formatter

Role required: admin

1. Navigate to **System UI > Formatters**.
2. Select **Parent Breadcrumbs**.
3. Click **View UI Macro for this Formatter** to view or modify the underlying formatter.

By default, the breadcrumb uses the default display field, `gr.getDisplayValue()`, as the link in the breadcrumb. To customize this, add the following line, replacing the `fieldName` parameter with the desired field name (not the field label):

```java
pc.setLabelField("fieldName")
```

If a user points to a breadcrumb, the short description for that record appears as a hint by default. To display alternate hover text for the breadcrumb, add the following line, replacing the `fieldName` parameter with the desired field name (not the field label):

```java
pc.setTitleField("fieldName")
```

Be sure to add these lines in the proper location, as shown in the following example:

```java
//parent crumb functions - script include
var pc = new ParentCrumbs(gr);
pcsetLabelField("short_description");

//override the default display field to be used for label
pc.setTitleField("number");
```
Use the parent breadcrumbs formatter on non-Task tables

The parent breadcrumbs formatter can be used on non-Task tables as long as the table has a reference to itself through a field called **parent**.

Role required: admin

To make the formatter available for a different table, duplicate the formatter used by the Task table:

1. Navigate to **System UI > Formatters**.
2. Select **Parent Breadcrumbs**.
3. Set the **Table** field to the appropriate table.
4. Right-click the form header and choose **Insert**.
5. Add the new formatter to the appropriate form.

Embed lists within a form

You can embed lists within a form. When a list is embedded in a form, any changes made to the contents of the list are saved when the form is saved.

Role required: personalize_form

This allows the embedded list to be treated just like another element on the form. In addition, users can create new records in the list view. After a row is added to the list, double-click any cell in the list to edit its value. You can add the same lists to a form as embedded lists or as related lists, depending on the path selected in the context menu.

You can also modify embedded list controls, such as the name of the related list and the roles required to create new records in the embedded list.

1. Select a record from the list.
2. Right-click the form header and select **Configure > Form Layout**.
   - The slushbucket shows the available fields and the lists that can be embedded in the form. Lists appear in red at the bottom of the **Available** list.
3. Select a list and move it to the **Selected** column.
4. Use the up and down arrow buttons to position the list in the form.
5. Click **Save**.
   - Lists adjust to fit the frames of the adjacent fields.
Secure records in an embedded list

To apply security to the records in embedded lists, limit editing and deleting records in embedded lists to specific roles.
Elevate to the security_admin role.

Role required: security_admin

1. Navigate to System Security > Access Control (ACL).
2. Open the Write or Delete record for the appropriate table.
3. In the Requires Role section of the form, add the roles that have write or delete permission for that table.
4. Save the changes.

When records from the associated table appear in an embedded list, the edit and delete options are available only to users with the specified roles.

Field administration

The individual pieces of data in a record are called fields. Users can enter data in fields by using the list editor or by using a form. In form view, fields appear as fields in the form, and in list view they appear as columns of data in the table. Administrators can create new or modify existing fields.

Field types

These field types are available to administrators when creating fields or changing the type of existing fields.

Note: If you edit a field on a child table that is present on the parent table, you also change it for the parent table and all other child tables.

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Field for uploading and embedding .mp3 or .ogg audio files.</td>
</tr>
<tr>
<td>Choice</td>
<td>List of choices that can be configured.</td>
</tr>
</tbody>
</table>
| Color         | String field that accepts CSS color declarations (including hex or RGB notation) and displays a preview. See:  
                   - HTML Colors (W3CSchools) for more information on hex and RGB notation.  
                   - HTML Color Names (W3CSchools) for valid color names.             |
<p>| Condition String | Text field that accepts a plain JavaScript condition statement that is validated automatically for correctness before an update. |
| Conditions    | Field that adds the condition builder to a form. Specify a dependent field that references the table name. |</p>
<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Currency</td>
<td>Decimal field with two digits after the decimal point. When displayed on a form, a currency field also includes an extra choice list for selecting the currency type. If there is no default value for the field, empty currency fields use the default system currency. Adding a value causes the field to use the preferred currency of the active user.</td>
</tr>
<tr>
<td>Data Structure</td>
<td>Field that allows the selection of one of the following data structures and entry of values to organize particular information in the record.</td>
</tr>
<tr>
<td>Date</td>
<td>Day, which can be selected with a calendar widget.</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Day and time of day, which can be selected with a calendar widget.</td>
</tr>
<tr>
<td>Decimal</td>
<td>Number with up to two digits after the decimal points (for example, 12.34).</td>
</tr>
<tr>
<td>Document ID</td>
<td>Reference to any field on any table.</td>
</tr>
<tr>
<td>Domain ID</td>
<td>System field that contains a reference to the domain.</td>
</tr>
<tr>
<td>Due Date</td>
<td>String input field that stores a date-time.</td>
</tr>
<tr>
<td>Duration</td>
<td>Length of time. Stored in the database as a positive integer number of milliseconds, but displays in days, hours, minutes, and seconds. Negative duration values are not supported.</td>
</tr>
<tr>
<td>Encrypted Text</td>
<td>Field that is encrypted for security. Depending on the user's encryption context, the field can be hidden, the value can be hidden, or the field and value can display.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Reference field for a field name in the table selected in a Table Name field type. Make this field dependent on the Table Name field.</td>
</tr>
<tr>
<td>Floating Point Number</td>
<td>Number with up to seven digits after the decimal point.</td>
</tr>
<tr>
<td>HTML</td>
<td>String field with a built-in HTML editor.</td>
</tr>
<tr>
<td>Image</td>
<td>Field for uploading and embedding images.</td>
</tr>
<tr>
<td>Integer</td>
<td>Number with zero decimal points.</td>
</tr>
<tr>
<td>Field Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Journal</td>
<td>Field that accepts text entries and displays previous entries with a user name and timestamp.</td>
</tr>
<tr>
<td>Journal Input</td>
<td>Field that accepts text entries but does not display previous entries.</td>
</tr>
<tr>
<td>Journal List</td>
<td>Field that displays the contents of journal fields. Specify the journal fields as the dependent fields. If a journal list field depends on more than one journal field, the entries are displayed chronologically.</td>
</tr>
<tr>
<td>List</td>
<td>Reference field that accepts multiple references rather than just one.</td>
</tr>
<tr>
<td>Long</td>
<td>Integer field that can contain a longer number than the integer field.</td>
</tr>
<tr>
<td>Password (1 Way Encrypted)</td>
<td>Text field that stores passwords with one-way encryption. One-way encryption stores the password as a secure hash value that cannot be decrypted.</td>
</tr>
<tr>
<td>Password (2 Way Encrypted)</td>
<td>Text field that stores passwords with two-way encryption. Two-way encryption stores the password as a secure encrypted value that can be decrypted programmatically within the instance. You can use Password 2 encryption with form variables. To encrypt text fields on forms, use Encryption Contexts.</td>
</tr>
<tr>
<td>Percent Complete</td>
<td>Decimal field that renders a percent complete bar in lists.</td>
</tr>
<tr>
<td>Phone Number (E164)</td>
<td>String field that provides E164-compliant formatting and validation for telephone numbers.</td>
</tr>
<tr>
<td>Price</td>
<td>Decimal field with an additional list for currencies.</td>
</tr>
<tr>
<td>Reference</td>
<td>Query that displays records from another table.</td>
</tr>
<tr>
<td>Script</td>
<td>Text field that accepts JavaScript code input and provides controls, such as syntax checking and formatting. It also provides a list of fields and server APIs. Specify a dependent field that references the table name for the list of fields.</td>
</tr>
<tr>
<td>Script (Plain)</td>
<td>Text field that accepts JavaScript code input and provides controls, such as syntax checking and formatting.</td>
</tr>
<tr>
<td>Field Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>String</td>
<td>For 254 characters or less, the string field is a single-line text field. Anything 255 characters or over appears as a multi-line text box. Note: The database translates the value you provide in the Max length field to the closest matching database field type. For example, a max string length of 80 maps to the nearest database field type of VARCHAR(100). For Oracle instances, users are not able to increase the max length of a string field to anything greater than 4000 through the application UI because the increase requires the CLOB datatype in Oracle. If you require a field to be larger than the max length, you can log an incident to request that a DBA change the max length for you.</td>
</tr>
<tr>
<td>String (Full UTF-8)</td>
<td>A string field that can contain UTF-8 character encoding. The string restriction on the number of characters applies to this field type.</td>
</tr>
<tr>
<td>Suggestion</td>
<td>String field that provides suggested values but accepts free-form text. Available when you add a field by configuring a form or list. Otherwise, you must modify the dictionary entry of an existing string or journal field.</td>
</tr>
<tr>
<td>Table Name</td>
<td>String field that lets you select a table. If you use the Field Name field type, add this field type and make the Field Name field dependent on the Table Name field.</td>
</tr>
<tr>
<td>Time</td>
<td>Specific time. Stored in the database as a string in milliseconds, but displays in hours, minutes, and seconds.</td>
</tr>
<tr>
<td>Translated HTML</td>
<td>HTML field that displays different translations based on the user's language.</td>
</tr>
<tr>
<td>Translated Text</td>
<td>Text field that displays different translations based on the user's language.</td>
</tr>
<tr>
<td>True/False</td>
<td>Boolean field that appears as a check box.</td>
</tr>
<tr>
<td>URL</td>
<td>String field that is a clickable URL field when locked.</td>
</tr>
<tr>
<td>Video</td>
<td>Field for uploading and embedding video.</td>
</tr>
<tr>
<td>Wiki</td>
<td>String field with a built-in Wikitext editor that accepts a simplified version of standard Wikitext formatting.</td>
</tr>
</tbody>
</table>
Modify string field length

You can modify the maximum character limit for a string field.

Role required: personalize_dictionary

1. Right-click the field label in the form and select Configure Dictionary.
2. Change the Max length field to the desired length.

   Note: You can change between string-based data types as long as length changes do not cause any data loss from truncation. For example, you can change from a MEDIUM database type to a VARCHAR(100) database type if none of the existing data is greater than 100.

3. Click Update.
   The system cancels any length change that results in data loss due to truncation.

Enable the text field character counter

By default, multi-line text fields have a 4000 character limit. To help users see how many characters remain before they reach the limit, you can enable the glide.ui.textarea.character_counter property.

Role required: admin

This property adds a counter under text fields, such as the Additional Comments and Work notes fields. The counter is dynamically updated as users enter text.

1. Navigate to System Properties > UI Properties.
2. Select the check box next to Character counter for textarea (journal and multi line text fields).
3. Click Save.
Valid data type changes

You can only change a dictionary entry's data type when the change does not result in data loss. Use the following guidelines to change a dictionary entry's data type.

Table 191: Valid data type changes

<table>
<thead>
<tr>
<th>Condition</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The field is empty in all table records.</td>
<td>None. You can convert an empty field from any data type to another without restriction.</td>
</tr>
<tr>
<td>The table contains existing data for the field.</td>
<td>You can only convert between logical data types that map to the same physical data type in the database. For example, you can convert a glide duration to a glide datetime since both logical data types map to the DATETIME physical data type in the database.</td>
</tr>
<tr>
<td>The field is a string field you are converting to another type of string field.</td>
<td>You can change between string-based data types as long as length changes do not cause any data loss from truncation. For example, you can change from a MEDIUM database type to a VARCHAR(100) database type if none of the existing data is greater than 100.</td>
</tr>
<tr>
<td>The field is a string field you are converting to a globally unique ID (GUID).</td>
<td>You can only convert a string field to a GUID if all of the existing data in the field are already Sys ID values.</td>
</tr>
<tr>
<td>The field is a GUID field you are converting to a string field.</td>
<td>None. You can convert a GUID field to a string field without restriction.</td>
</tr>
</tbody>
</table>

Database field types

Available field types and their corresponding MySQL database types.

Typically, it isn't necessary to perform any actions at the database level. To learn about changing a field type, see System dictionary on page 1443.

Table 192: Database field types

<table>
<thead>
<tr>
<th>Field types</th>
<th>Options</th>
<th>Dictionary XML type</th>
<th>MySQL DB type</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>small</td>
<td>string</td>
<td>VARCHAR(40)</td>
</tr>
<tr>
<td>String</td>
<td>medium</td>
<td>string</td>
<td>VARCHAR(100)</td>
</tr>
<tr>
<td>String</td>
<td>large</td>
<td>string</td>
<td>MEDIUMTEXT</td>
</tr>
<tr>
<td>String</td>
<td>extralarge</td>
<td>string</td>
<td>MEDIUMTEXT</td>
</tr>
<tr>
<td>Decimal</td>
<td></td>
<td>decimal</td>
<td>Decimal (15,2) older instances may have (12,2)</td>
</tr>
<tr>
<td>Integer</td>
<td></td>
<td>integer</td>
<td>Integer</td>
</tr>
<tr>
<td>True-False</td>
<td></td>
<td>boolean</td>
<td>TINYINT(1)</td>
</tr>
</tbody>
</table>
Add a new field to a table

Administrators can add new fields to a table to store and display data.

Role required: admin

**Warning:** Do not add more than 10 medium-length or longer String fields to a single table. Attempting to save a large number of characters in 11 or more String fields can result in the following error: Syntax Error or Access Rule Violation detected by database (Row size too large (> 8126)).

1. Navigate to any form.
2. Right-click the form header and select **Configure > Form Layout**.
3. In the **Create new field** section, fill in the following fields:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the field as you want it to appear on forms and lists.</td>
</tr>
<tr>
<td>Database column name</td>
<td>Enter the database name for the field.</td>
</tr>
<tr>
<td>Type</td>
<td>Select a field type.</td>
</tr>
<tr>
<td>Field length</td>
<td>Select a field length. This field is visible only for certain field types.</td>
</tr>
</tbody>
</table>

4. Click **Add**.
5. Use the slushbucket to place the field in the desired location on the form.
6. Click **Save**.
   The field now appears on the form in the designated location.

---

<table>
<thead>
<tr>
<th>Field types</th>
<th>Options</th>
<th>Dictionary XML type</th>
<th>MySQL DB type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>glide_date</td>
<td>glide_date</td>
<td>DATE</td>
</tr>
<tr>
<td>Date-Time</td>
<td>glide_date_time</td>
<td>DATETIME</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>glide_time</td>
<td>DATETIME</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td>glide_duration</td>
<td>DATETIME</td>
<td></td>
</tr>
<tr>
<td>Choice</td>
<td>string</td>
<td>VARCHAR(40)</td>
<td></td>
</tr>
<tr>
<td>Suggestion</td>
<td>string</td>
<td>VARCHAR(40)</td>
<td></td>
</tr>
<tr>
<td>Journal</td>
<td>journal</td>
<td>MEDIUMTEXT</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td>&lt;reference table&gt;</td>
<td>reference</td>
<td>VARCHAR(32)</td>
</tr>
<tr>
<td>List</td>
<td>glide_list</td>
<td>MEDIUMTEXT</td>
<td></td>
</tr>
<tr>
<td>Url</td>
<td>url</td>
<td>MEDIUMTEXT</td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>user_image</td>
<td>VARCHAR(40)</td>
<td></td>
</tr>
<tr>
<td>Due-Date</td>
<td>due_date</td>
<td>DATETIME</td>
<td></td>
</tr>
</tbody>
</table>
Make a field mandatory

Fields can be marked as mandatory, meaning they must contain a value before the record can be saved. Mandatory fields are marked with a field status indicator before the label.

Role required: personalize_dictionary

A mandatory field that populated by the platform with default data, such as a value from a client script, does not display the indicator. If you delete this value from the field, however, the indicator appears. The color of the indicator depends on the field state. For more information, see UI16 and UI15 field status indicators on page 106 or UI11 field status indicators on page 106.

1. Right-click the field's label in the form and select Configure Dictionary.
2. In the Dictionary form, select the Mandatory check box.
3. Click Update.

The next time the form is opened, a field status indicator appears next to the field label, indicating that a value is mandatory.

**Note:** Mandatory fields are global. The field is marked as mandatory everywhere it appears in a form.

**Mandatory Reference Fields**

A form can be saved with an empty mandatory field, if that field is a reference field (derived from another table) and if the parent field is also blank. However, if the mandatory reference field shows a value from the parent field, then the form cannot be saved if this value is deleted. If the value in the referenced field is changed, the value for that field is changed everywhere it appears.

Specify a default field value

A default value populates a value in a field when a new record is created.

Role required: personalize_dictionary

The default value populates the field on the blank form for a new record, and also when the new record is submitted if the field is empty. Default values can be specified as either a constant or generated through script.

1. Right-click the field's label in the form and select Configure Dictionary.
2. Enter the default value in the Default value field.

To set a constant value, type it into the Default value text box field. To assign a default value using a more complex formula, use JavaScript to output a default value.

**Note:** The default value is the underlying value that would be present in the field, not the label. For example, in a choice list field, use the value of the choice as the default value, not the choice's name.

**Constant default values**

**Javascript default values**

**Set a default value for assignment_group**

**Set a default value for assigned_to if user has the itil role**

**Set a default value for a duration field**

Set a default value for the Priority field.
To view out-of-box examples of JavaScript default values, navigate to **System Definition > Dictionary** and enter this filter: `[Default value] [starts with] [javascript]`. Open some of the records and view the default value javascript entries.

The following example sets a default value in a `[sys_user_group]` reference field by getting the ID from the name of a group.

```
javascript:GetIDValue('sys_user_group', 'Development');
```

Configure the default value for the `[assigned_to]` user equal to the current user id if the user has a role of itil.

- Column label: **Assigned to**
- Column name: `[assigned_to]`
- Reference Specification section
  - Reference: **User [sys_user]**
  - Reference qual condition: `[Roles] [is] [itil]`
- Default value script:

```javascript
javascript:if (gs.hasRole("itil")) current.assigned_to = gs.getUserID();
```

To set a default value for a duration field, use the following in the Default value field of the duration field's dictionary entry:

```javascript
javascript:current.duration_field.setDisplayValue('3 04:30:14');
```

Do not hard-code a particular date-time because the value becomes invalid if the system date-time format changes.

Make a field dependent

A choice or reference field can be declared dependent on another field on the same table. Dependent fields limit their available values based on the value in the dependent field.

Role required: personalize_dictionary

If a required dependency does not function as expected, as might happen if there is a many-to-many relationship between the fields, consider using reference qualifiers to accomplish the goal.

1. Right-click the field label in the form and select Configure Dictionary.
2. In theDependentfield, enter the name of the field that this field will depend on.
3. Click Update when done.

The "subcategory" field is made dependent on "category". The "category" value in a form will determine which options appear for the "subcategory" field.

**Note:** Fields cannot be made dependent on derived fields.

Require unique values for a field

You can require that a field's values are unique. This means no two records have the same value for that field.

Role required: personalize_dictionary
Warning: Making a field unique when the corresponding table already has different values for that field causes data loss. All duplicate records are deleted.

By default, fields are created without this constraint. A field can have unique values only if there are not already duplicate values in the database for that field.

1. Verify that no records in the table for the field have values, or that they all have the same value.
2. Right-click the field label in the form and select Configure Dictionary.
3. Configure the form to add the Unique field if it does not already appear.
4. Select the Unique check box.
5. Update.

Change the field label or hint

You can change a field’s label or the text that appears as a hint when you point your cursor to the field.

Role required: personalize_dictionary

1. Navigate to the form the field appears on.
2. Right-click the field label and select Configure Label.
3. Update the form.

Table 193: Changing the Field Label or Hint

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>The table the label appears on.</td>
</tr>
<tr>
<td>Label</td>
<td>The label that the field displays. You can enter up to 50 characters for the label, but not all databases support this many characters. The recommended maximum is 30 or fewer characters. HTML in field labels is not supported.</td>
</tr>
<tr>
<td>Plural</td>
<td>The plural version of the field label.</td>
</tr>
<tr>
<td>Element</td>
<td>The dictionary name of the field.</td>
</tr>
<tr>
<td>Help</td>
<td>Helpful text stored in the record.</td>
</tr>
<tr>
<td>Hint</td>
<td>A short description of the field that displays when the user points to it.</td>
</tr>
<tr>
<td>URL</td>
<td>If this field is not blank, a URL link that displays on the label.</td>
</tr>
<tr>
<td>URL target</td>
<td>A target attribute that determines where the URL opens. For information on the target attributes, see this W3Schools article.</td>
</tr>
</tbody>
</table>

4. Click Update.
Delete fields

You can delete custom fields that you created. Custom fields begin with u_. Remove the field from forms and lists instead of deleting it.

Role required: admin

You cannot delete base system fields. In addition, any missing base system fields are recreated when the instance is upgraded.

1. Navigate to a form that contains a custom field to delete.
2. Right-click the field label and select Configure Dictionary.
3. Click Delete Column in the form header, and then click OK.
4. To delete multiple custom fields, complete the following steps.
   a) Navigate to System Definition > Dictionary.
   b) Locate the custom fields to delete.
      For example, search for column names that start with u_.
   c) Check the boxes next to the fields to delete and select Delete from the action list below the list.
      A confirmation dialog opens. If there are dependencies for the selected fields, they are listed.
   d) Click Delete.

UI16 and UI15 field status indicators

Indicators are used on some fields to denote a special field type.

A field status indicator is a colored asterisk that may appear to the left of mandatory fields. Field status indicators change colors to represent different states of mandatory fields.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unpopulated</td>
<td>Required field is empty. The user must enter a value to save the form. Default color is red.</td>
</tr>
<tr>
<td>Populated - saved</td>
<td>Required field contains a value that was saved or needs to be saved. Default color is grey.</td>
</tr>
<tr>
<td>Populated - unsaved</td>
<td></td>
</tr>
</tbody>
</table>
Figure 247: UI16/UI15 field status indicators

Note: Tabs containing required fields have black asterisks.

UI11 field status indicators

A field status indicator is a colored bar that may appear to the left of form elements. These indicators convey the following statuses.

<table>
<thead>
<tr>
<th>Field status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Required field that is empty. The user must enter a value to save the form. Default color is red.</td>
</tr>
<tr>
<td>Populated mandatory</td>
<td>Required field for which a value has already been saved. Default color is light red. If the user enters a new value, the field status indicator changes to modified (default green).</td>
</tr>
<tr>
<td>Modified</td>
<td>Contains data that has not been saved. Default color is green.</td>
</tr>
<tr>
<td>Read-only</td>
<td>User cannot edit on the form. Default color is orange.</td>
</tr>
</tbody>
</table>
Figure 248: UI11 field status indicators

An administrator can customize the color of these field status indicators by navigating to **System Properties > CSS**:

Figure 249: Customize UI11 field status indicator colors

**HTML fields**

An HTML field allows users to define how field content is rendered by using HTML. Knowledge articles, service catalog item descriptions, release documentation, and HTML content blocks are common examples of HTML fields. Administrators can **add HTML fields** to any form and also **customize the functionality** of HTML fields.

**Types of HTML editors**

There are two types of HTML editors in the system.

- TinyMCE: A What You See Is What You Get (WYSIWYG) field that displays text as readers would see it on the screen. TinyMCE is the default editor.
- htmlArea: The legacy editor, which offers a more basic WYSIWYG interface as well as a mode that shows only HTML markup.

The **glide.ui.html.editor** system property specifies which interface is used for all HTML fields.
Note: Styles declared by HTML fields may be overridden by CSS declarations in the themes.css file that styles the entire instance. To determine what attributes are overriding a style, use a web development tool (for example, Firebug for Firefox).

TinyMCE HTML editor

Versions of the TinyMCE editor that are available.

- Version 4, which provides a simplified toolbar and an updated table designer tool.
- Version 3.

The version of the TinyMCE editor currently in use depends on the user interface:

- The UI16 and UI15 interfaces use version 4.
- The UI11 interface uses version 3.

Figure 250: TinyMCE v4
Figure 251: TinyMCE v3

Users can modify the TinyMCE toolbar and add or remove functionality. For more information, see Configure the TinyMCE HTML toolbar on page 809.

HtmlArea editor

The HtmlArea editor is the legacy editor for HTML fields.

Figure 252: HtmlArea

Use the editors

The editors provide controls that are similar to a word processing program.

Formatting
The formatting table displays how to control the way text appears.

In the following table, the TinyMCE v3/htmlArea icon column displays icons that are available with the TinyMCE version 3 and htmlArea editors. The TinyMCE v4 icon column displays icons that are available with the TinyMCE version 4 editor. Names marked with an asterisk (*) are not available with the htmlArea editor.
### Table 195: Formatting table

<table>
<thead>
<tr>
<th>TinyMCE v3/htmlArea icon</th>
<th>TinyMCE v4 Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✒️</td>
<td>New Document*</td>
<td>Clears the contents of the HTML field.</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>Bold</td>
<td>Applies bold formatting to the selected text or current word. Keyboard shortcut: CTRL + B</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>Italic</td>
<td>Applies italics formatting to the selected text or current word. Keyboard shortcut: CTRL + I</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>Underline</td>
<td>Applies underline formatting to the selected text or current word. Keyboard shortcut: CTRL + U</td>
</tr>
<tr>
<td>ABC</td>
<td></td>
<td>Strikethrough*</td>
<td>Applies strikethrough formatting to the selected text or current word.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Align Right</td>
<td>Applies right alignment to the current paragraph.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em>Code reference: text-align:right</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Align Center</td>
<td>Applies center alignment to the current paragraph.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em>Code reference: text-align:center</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Align Left</td>
<td>Applies left alignment to the current paragraph.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em>Code reference: text-align:left</em></td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved. 793
<table>
<thead>
<tr>
<th>TinyMCE v3/htmlArea icon</th>
<th>TinyMCE v4 icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Justify</td>
<td>Applies justified alignment, which stretches the lines to equal width, to the current paragraph.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em>Code reference:</em> <code>text-align: justify</code></td>
</tr>
<tr>
<td>Paragraph</td>
<td></td>
<td>Format</td>
<td>Applies a paragraph style to the current paragraph, such as Paragraph, Heading 1, and Preformatted.</td>
</tr>
<tr>
<td>Font Family</td>
<td></td>
<td>Font Family</td>
<td>Applies a font family to the selected text or current word.</td>
</tr>
<tr>
<td>Arial</td>
<td></td>
<td>Font Sizes</td>
<td>Applies a font size to the selected text or current word.</td>
</tr>
<tr>
<td>3 (12pt)</td>
<td></td>
<td>Insert/Remove Bulleted List</td>
<td>Applies or removes unordered list tags for the selected paragraphs. Click the arrow beside the button to select a different bullet type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Insert/Remove Numbered List</td>
<td>Applies or removes ordered list tags for the selected paragraphs. Click the arrow beside the button to select a different number type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decrease Indent</td>
<td>Removes indentation from the current or selected paragraphs (removes 30px of left padding; padding cannot be less than 0).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em>Code reference:</em> <code>padding-left</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Increase Indent</td>
<td>Applies indentation to the current or selected paragraphs (adds 30px of left padding).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em>Code reference:</em> <code>padding-left</code></td>
</tr>
<tr>
<td>TinyMCE v3/htmlArea Icon</td>
<td>TinyMCE v4 Icon</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| “                       |                | Block Quote* | Applies the `<blockquote>` tag, which defines a long quotation, to the current or selected paragraphs. Browsers usually indent these elements.  
**Code reference:** `<blockquote>` |
| A_ _                     | A_ _           | Select Text Color | Applies font color to the current word or selected text. Click the button to use the current color, or click the arrow next to the button to view more colors. Click More Colors... to view various color options and the hexadecimal codes. |
| ab                       | A_ _           | Select Background Color | Applies background color to the current word or selected text. Click the button to use the current color, or click the arrow next to the button to view more colors. Click More Colors... to view various color options and the hexadecimal codes. |
|                          |                | Clear Formatting* | Removes the inline styles and formatting from the selected text. |
| x_2                      |                | Subscript* | Applies subscript text, which appears half a character below the baseline, to the current word or selected text.  
**Code reference:** `<sub>` |
**Table functions in the TinyMCE version 4 editor**

The TinyMCE version 4 editor uses menus and menu selections to create and edit tables.

<table>
<thead>
<tr>
<th>UI element</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Table Icon" /></td>
<td>Click the table icon ( ) to access the TinyMCE version 4 table menu. Use the table menu to:</td>
</tr>
<tr>
<td><img src="image" alt="Insert Table" /></td>
<td>- Insert or delete a table</td>
</tr>
<tr>
<td><img src="image" alt="Table Properties" /></td>
<td>- Modify table properties</td>
</tr>
<tr>
<td><img src="image" alt="Delete Table" /></td>
<td>- Add, move, or delete rows and columns</td>
</tr>
<tr>
<td><img src="image" alt="Cell" /></td>
<td>- Modify row and column properties</td>
</tr>
<tr>
<td><img src="image" alt="Row" /></td>
<td>- Split and merge cells</td>
</tr>
</tbody>
</table>

**Table 196: TinyMCE version 4 editor**

<table>
<thead>
<tr>
<th>TinyMCE v3/htmlArea icon</th>
<th>TinyMCE v4 icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Superscript Icon](image) | | Superscript* | Applies superscript text, which appears half a character above the baseline, to the current word or selected text. *

*Code reference:* `<sup>`
To insert a table in the HTML field, click **Insert table** and highlight squares in the grid to represent the desired number of rows and columns. Click the last highlighted square to insert the table.

After you insert the table, you can modify the size by clicking and dragging the handles at the table edges.

From the table menu, click **Table properties** to open the Table properties dialog box. From this box you can take any of the following actions.

- **General tab:**
  - Set table width and height
  - Set cell spacing and padding
  - Enable borders and captions
  - Set the table alignment

- **Advanced tab:**
  - Configure the table style
  - Select the border color
  - Select the background color
With the cursor in the desired table cell, open the table menu and click **Cell properties**. From this box you can set the following properties for table cells.

- **General tab:**
  - Width and height
  - Type and scope
  - Horizontal and vertical alignment

- **Advanced tab**
  - Configure the cell style
  - Select the border color
  - Select the background color

With the cursor in a table cell in the desired row, open the table menu and click **Row properties**. From this box you can set the following properties for rows.

- **Row type**
- **Alignment**
- **Height**

- **Advanced tab**
  - Configure the row style
  - Select the border color
  - Select the background color

---

**Table functions in the TinyMCE version 3 and htmlArea editors**

The TinyMCE version 3 editor and the htmlArea editor use icons to create and edit tables.

Names marked with an asterisk (*) are not available with the htmlArea editor.
<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![insert/edit table icon] | Insert/Edit Table           | Inserts a table and defines properties for the current table, including columns, rows, width, layout, and spacing. To learn more, see the table styles example.  
*Code reference: table* |
| ![table row properties icon] | Table Row Properties*       | Defines properties for the current row, odd rows, even rows, or all rows in the table. To learn more, see the table styles example.  
*Code reference: tr* |
| ![table cell properties icon] | Table Cell Properties*      | Defines properties for the current cell, cells in the current row, cells in the current column, or all cells in the table. To learn more, see the table styles example.  
*Code reference: td* |
| ![insert row before icon]   | Insert Row Before*          | Adds a row above the current row in a table.                                                                                                 |
| ![insert row after icon]    | Insert Row After*           | Adds a row below the current row in a table.                                                                                                 |
| ![delete row icon]          | Delete Row*                 | Deletes the current row in a table.                                                                                                           |
| ![insert column before icon]| Insert Column Before*       | Adds a column to the left of the current column in a table.                                                                                   |
| ![insert column after icon] | Insert Column After*        | Adds a column to the right of the current column in a table.                                                                                   |
| ![delete column icon]       | Delete Column*              | Deletes the current column in a table.                                                                                                         |
| ![split merged table cells icon] | Split Merged Table Cells*  | Splits any merged cells in the selected table cells.                                                                                          |
### Editing functions

The options you can use to edit HTML content.

#### Table 198: Editing functions

<table>
<thead>
<tr>
<th>Icon</th>
<th>TinyMCE v4 Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="cut-icon.png" alt="Cut" /></td>
<td><img src="cut-icon.png" alt="Cut" /></td>
<td>Cut*</td>
<td>Cuts the selected text. Not supported in all browsers; use keyboard shortcut. Keyboard shortcut: CTRL + X</td>
</tr>
<tr>
<td><img src="copy-icon.png" alt="Copy" /></td>
<td><img src="copy-icon.png" alt="Copy" /></td>
<td>Copy*</td>
<td>Copies the selected text. Not supported in all browsers; use keyboard shortcut. Keyboard shortcut: CTRL + C</td>
</tr>
<tr>
<td><img src="paste-icon.png" alt="Paste" /></td>
<td><img src="paste-icon.png" alt="Paste" /></td>
<td>Paste*</td>
<td>Pastes the selected text. Not supported in all browsers; use keyboard shortcut. Keyboard shortcut: CTRL + V</td>
</tr>
<tr>
<td><img src="plain-text-icon.png" alt="Paste as Plain Text" /></td>
<td><img src="plain-text-icon.png" alt="Paste as Plain Text" /></td>
<td>Paste as Plain Text*</td>
<td>Enables paste as plain text without source formatting.</td>
</tr>
<tr>
<td><img src="paste-word-icon.png" alt="Paste from Word" /></td>
<td><img src="paste-word-icon.png" alt="Paste from Word" /></td>
<td>Paste from Word*</td>
<td>Opens a new window that allows you to copy and paste content from Microsoft Word into the HTML field.</td>
</tr>
<tr>
<td><img src="find-icon.png" alt="Find" /></td>
<td><img src="find-icon.png" alt="Find" /></td>
<td>Find*</td>
<td>Allows you to locate text strings in the HTML field. Search above (up) or below (down) the cursor location.</td>
</tr>
</tbody>
</table>
**Table 199: Extended functions**

<table>
<thead>
<tr>
<th>Icon</th>
<th>TinyMCE v4 Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Find/Replace Icon" /></td>
<td></td>
<td>Find/Replace*</td>
<td>Allows you to replace the next (Replace) or all (Replace All) occurrences of a text string in the HTML field.</td>
</tr>
<tr>
<td><img src="image" alt="Undo Icon" /></td>
<td></td>
<td>Undo*</td>
<td>Reverts the previous edit.</td>
</tr>
<tr>
<td><img src="image" alt="Redo Icon" /></td>
<td></td>
<td>Redo*</td>
<td>Reapplies the last reverted edit.</td>
</tr>
</tbody>
</table>

*These options are not available with htmlArea.

*Extended functions*

The extended functions available for working with HTML content.

<table>
<thead>
<tr>
<th>Icon</th>
<th>TinyMCE v4 Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Insert/Edit Link Icon](image) | | Insert/Edit Link | Configures a link for the selected text. Define the link URL, title (additional information that appears in the tool tip), and the target (same window or new window or tab).  

*Code reference:* `<a>` |
<p>| <img src="image" alt="Remove Link Icon" /> | | Remove link* | Removes the current hyperlink. |
| <img src="image" alt="Cleanup Messy Code Icon" /> | | Cleanup Messy Code* | Fixes standard HTML errors for the selected text, such as invalid tags. Note that clicking this button may change the layout of existing content. You can click <strong>Undo</strong> to revert this action if you do not like the results. |
| <img src="image" alt="Edit HTML Source Icon" /> | | Edit HTML Source | Opens HTML source code in a separate window. See <em>Editing in HTML Source Mode.</em> |</p>
<table>
<thead>
<tr>
<th>Icon</th>
<th>TinyMCE v4 Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Insert Horizontal Line</td>
<td>Inserts a horizontal line at the current location.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Toggle Invisible Elements*</td>
<td>Shows or hides invisible elements in the article, such as collapsed table borders.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Insert Special Character*</td>
<td>Inserts a special character (symbol) at the current cursor location. Click the button to view a list of available characters. Point to a character to view the name and HTML code. Click a character to insert it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Insert/Edit Image</td>
<td>Inserts an image from the image library or an attachment. You can also add images to the image library with this feature. To learn more, see <a href="#">Embedding Images in HTML Fields</a>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Insert/Edit Embedded Media</td>
<td>Embeds a video from the video library or an attachment. You can also add videos to the video library with this feature.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spell Checker</td>
<td>Checks the spelling of text in the HTML field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preview*</td>
<td>Opens a preview of the HTML field in a separate window without saving changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Toggle Full Screen Mode*</td>
<td>Expands the HTML field to use the full form view for easier editing. Click the button again to return to standard form view. This feature is not available for Internet Explorer.</td>
</tr>
</tbody>
</table>
*These options are not available with htmlArea.

**Highlight text in the TinyMCE editor**

On the bottom bar of the TinyMCE editor, the path of HTML tags for the text at the cursor position is displayed.

Click a tag in the path to highlight the text affected by the tag.

**Insert a line break in the HTML editor**

When you use the Enter key, the editor creates a new paragraph element (<p>) tag, which appears as a double space.

To enter a single line space, use the Shift + Enter key combination, which inserts a line break (<br>) tag.

**Use HTML source mode in the htmlArea editor**

In HTML source mode, you can use standard HTML to edit text.

Click the HTML source icon (HTML) to open the HTML source code in a separate window. Make your changes and then click **Update**.

![HTML Source Editor](image)

**Figure 253: HTML editor**

**Note:** Fields which are blank may still have default HTML tags applied by default - for example, a <body></body> tag. These tags will be displayed in the HTML Source Mode.

**Table style example**

This example uses HTML field controls to format a table in a knowledge article.
To add the formatted table to a knowledge article:

1. Navigate to **Knowledge > Edit** and select the article to edit.
2. In the HTML field, position the cursor in the location for the table.
3. Click the table icon, click **Insert table**, and then select the number of rows and columns.
4. Complete the following steps to edit the table properties.
   a) Position your cursor in the table, click the table icon, and select **Table properties**.
   b) Enter the following values on the **General** tab.
      • Width: 75%
      • Cell spacing: 3
      • Cell padding: 3
      • Border: 1
      • Alignment: Left
   c) On the **Advanced** tab, click the text field next to **Border color** and enter **Gray**. The color picker box to the right turns gray to indicate the color you entered. You can also click the box and select the color in the palette.
   d) Click **Ok**.

5. Complete the following steps to update the header table row.
   a) Select the cells in the first table row, click the table icon, and select **Row > Row properties**.
   b) Enter the following values on the **General** tab.
      • Row type: Header
      • Alignment: Center
   c) On the **Advanced tab**, enter **#87cefa** in the text box beside **Background color** to set it to a light blue.
   d) Click **Ok**.

6. To set cell properties, complete the following steps.
   a) Select all table cells in the first column except those in the header row, click the table icon, and select **Cell > Cell properties**.
   b) Enter the following values on the **General** tab.
7. To set the background color of the middle row, complete the following steps.
   a) Position your cursor in the middle table row, click the table icon, and select **Row > Row properties**.
   b) On the **Advanced tab**, enter **Silver** in the text box beside **Background color** to set it to color #c0c0c0.
   c) Click **Ok**.

   You need to repeat this procedure for every other table row.

8. To set column width, complete the following steps.
   a) Click the first column of the table, click the table icon, and select **Cell > Cell properties**.
   b) On the **General tab**, enter **30%** in the **Width** text field.
   c) Click **Ok**.

9. Right-click the form header and click **Save**.
10. Enter data in the table cells and then save the article.

Adding media to HTML fields

Users can add media elements, including images and web links, to HTML fields.

Embedding videos is not available for this release due to browser plugin constraints. You can attach any video file to the form using the attachment option. For more information on attachments, see **Add attachments** on page 127.

**Embed images in HTML fields**

You can use the HTML field image picker to embed images into HTML fields, and to add images to the image library.

Role required: the role necessary to update the record that contains the HTML field. For example, any user with a role can create a knowledge article and embed an image in it.

**Note:** Administrators and users with the image_admin role manage the image library at **System UI > Images**. See **Storing images in the database** on page 205.

1. Open the form that contains the HTML field.
2. Click at the position where the image is to appear, or to modify an existing image, click the image.
3. Click the insert/edit image icon on the HTML editor toolbar.
   - The Insert/Modify Image form opens.
4. Enter information in each field.
### Table 200: Insert/Modify Image form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Type** | Select the image type.  
  - **Image Library**: may be reused.  
  - **Attachment**: available in the current record only |
| **Image** | Begin typing a file name and select an image from the list, or click the reference lookup icon and select an image.  
  To upload a new image, click **New**, click **Choose File**, locate the image, and click **Upload**.  
  If you chose the **Attachment** type, click **Choose File**, locate the image, and click **Attach**. |
| **Tooltip** | Enter alternate text that appears when a user points to the image. |
| **Alt** | Enter alternate text that can be used to improve accessibility. For example, it could be used with a screen reader. If this field is left blank, it defaults to the text entered in the **Tooltip** field. |

**Note:** To resize an embedded image (not available for Chrome browsers), click the image. The sizing frame appears. Drag a sizing point until the image is the desired size. Corner points adjust the size proportionally.

5. Optional: To provide additional control over the image's appearance, click **Advanced options**.

#### Table 201: Advanced options

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Layout</strong></td>
<td>Select the image <strong>Alignment</strong> (default is <strong>Baseline</strong>) and enter the <strong>Border thickness</strong>.</td>
</tr>
<tr>
<td><strong>Spacing</strong></td>
<td>Enter the number of <strong>Horizontal</strong> and <strong>Vertical</strong> pixels around the image.</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>Enter the <strong>Width</strong> and <strong>Height</strong> of the image (in pixels).</td>
</tr>
</tbody>
</table>

6. Click **OK**.

*Link to a website in HTML fields*  
You can insert a link to a website in an HTML field:
Role required: the role necessary to update the record that contains the HTML field. For example, any user with a role can create a knowledge article and link to a website in the article text.

1. Move the cursor to the position where the link is to appear.
2. Click the insert/edit link icon (🔗) in the HTML toolbar.
3. Enter information in each field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Enter the URL for the link. Copying and pasting is usually the easiest method.</td>
</tr>
<tr>
<td>Text</td>
<td>Enter the text you want to display for the link.</td>
</tr>
<tr>
<td>Target</td>
<td>Select the target window for the URL. For files, the None (use implicit) selection is generally the best choice. However, if you are linking to a complete web page, choose New window (_blank) so the browser opens the link in a new tab or window.</td>
</tr>
</tbody>
</table>

4. Click **OK** to insert the link into the field.

HTML field administration

An HTML field allows users to use HTML to define how field content is rendered.

The HTML editor provides WYSIWYG (what you see is what you get) functionality and HTML source mode editing. Administrators can customize some of the functionality associated with HTML fields. The HTML editors available depend on your version of the UI.

<table>
<thead>
<tr>
<th>HTML editor</th>
<th>UI support</th>
</tr>
</thead>
<tbody>
<tr>
<td>TinyMCE version 4</td>
<td>UI16, UI15</td>
</tr>
<tr>
<td>TinyMCE version 3</td>
<td>UI11</td>
</tr>
<tr>
<td>htmlArea (legacy)</td>
<td>UI11</td>
</tr>
</tbody>
</table>

Configure the TinyMCE HTML field editor

You can configure HTML fields to use the TinyMCE editor. This editor provides better stability and more editing functions than the legacy htmlArea editor.

Role required: admin

1. Navigate to **System Properties > UI Properties**.
2. Locate the property called **HTML field editor to use** (glide.ui.html.editor).
3. Select **TinyMCE**.
4. Click **Save**.
Configure the TinyMCE HTML toolbar

When HTML fields are configured to use the TinyMCE HTML editor, follow this procedure to configure which buttons are available on the toolbar.

Role required: admin

This procedure applies to both versions of the TinyMCE editor: version 3, available in UI11, and version 4, available in UI16 and UI15.

1. **Navigate to System Properties > UI Properties.**
2. **Locate the appropriate properties based on your version of the UI.**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| **UI16 or UI15**                                      | • (TinyMCE v.4.0.26) Configures the editing toolbar (first line) for HTML fields (glide.ui.html.editor.v4.toolbar.line1)  
• (TinyMCE v.4.0.26) Configures the editing toolbar (second line) for HTML fields (glide.ui.html.editor.v4.toolbar.line2) |
| **UI11**                                             | • Configures the editing toolbar (first line) for HTML fields (glide.ui.html.editor.toolbar.line1)  
• Configures the editing toolbar (second line) for HTML fields (glide.ui.html.editor.toolbar.line2) |

3. **Enter or remove buttons for each toolbar as a comma-separated list without spaces.**
   - Use a vertical bar ("|") to add a section separator.
   - Use the following button names when defining these properties. For a description of button functionality, see *Use the editors* on page 792.

<table>
<thead>
<tr>
<th>Button purpose</th>
<th>Accepted button names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formatting</td>
<td>newdocument, bold, italic, underline, strikethrough, justifyleft, justifycenter, justifyright, justifyfull, formatselect, fontselect, fontsizeselect, bullist, numlist, outdent, indent, blockquote, forecolor, removeformat, backcolor, sub, sup</td>
</tr>
<tr>
<td>Table functions</td>
<td>tablecontrols</td>
</tr>
<tr>
<td>Editing</td>
<td>cut, copy, paste, pastetext, search, replace, undo, redo</td>
</tr>
<tr>
<td>Extended functions</td>
<td>link, unlink, cleanup, code, hr, visualaid, charmap, image, media, preview, spellchecker, fullscreen (not supported by Internet Explorer)</td>
</tr>
</tbody>
</table>

The **spellchecker** tool is not supported in UI16 or UI15.

4. **Click Save.**
Configure TinyMCE editor to allow deprecated tags

You can set a dictionary attribute on a TinyMCE editor field to allow the use of deprecated HTML tags, such as `<b>` and `<i>`. By default, the TinyMCE editor uses the `<strong>` and `<em>` tags for bold and italic formatting.

Role required: personalize_dictionary or admin

After you set the dictionary attribute, use code view to manually enter deprecated tags. The editor does not validate any tags you enter manually, for example, if you type an incorrect character.

1. Navigate to the form with an HTML field that uses the TinyMCE editor.
2. Right-click the HTML field label and select **Configure dictionary**.

3. In the **Attributes** field, enter `tinymce_allow_all=true`, separated by a comma if needed. If other attributes are already listed, use a comma as a separator.
Configure the legacy HTML field editor
Administrators can configure HTML fields to use the legacy htmlArea editor in UI11.

Role required: admin

Note that the htmlArea HTML editor is not supported in UI16 or UI15.

1. Navigate to **System Properties** > **UI Properties**.
2. Locate the property called **HTML field editor to use** (glide.ui.html.editor).
3. Select **htmlArea**.
4. Click **Save**.

Configure the legacy HTML toolbar
When HTML fields are configured to use the legacy htmlArea HTML editor, follow this procedure to configure which buttons are available on the toolbar.

Note that the legacy HTML toolbar is not supported in UI16 or UI15.

1. Navigate to **System Properties** > **UI Properties**.
2. Locate the property called **Configures the editing toolbar for HTML fields** (glide.ui.html.toolbar).
3. Enter a comma-separated list, without spaces, to define the buttons that should appear on the toolbar. Use the following button names. For a description of button functionality, see **Use the editors** on page 792.
<table>
<thead>
<tr>
<th>Button purpose</th>
<th>Accepted button names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formatting</td>
<td>formatblock, fontname, fontsize, bold, italic, underline, justifyleft, justifycenter, justifyright, justifyfull, insertorderedlist, insertunorderedlist, outdent, indent, forecolor, hilitecolor</td>
</tr>
<tr>
<td>Editing</td>
<td>copy, paste, undo</td>
</tr>
<tr>
<td>Extended functions</td>
<td>createlink, inserthorizontalrule, insertimage, insertvideo, inserttable, htmlmode</td>
</tr>
</tbody>
</table>

4. Click **Save**.

**Note:**

The **Configures the editing toolbar (first line) for HTML fields** (glide.ui.html.editor.toolbar.line1) and **Configures the editing toolbar (second line) for HTML fields** (glide.ui.html.editor.toolbar.line2) properties only apply to the TinyMCE version 2 editor. The **Configures the editing toolbar (first line) for HTML fields** (glide.ui.html.editor.v4.toolbar.line1) and **Configures the editing toolbar (second line) for HTML fields** (glide.ui.html.editor.v4.toolbar.line2) properties only apply to the TinyMCE version 4 editor. Changes to these properties only apply when the TinyMCE editor is enabled; they do not affect the legacy htmlArea editor.

---

**Components installed with the TinyMCE HTML field editor**

The following components are included with the TinyMCE HTML Field Editor plugin.

- Properties
- Business rules

**Properties installed with the TinyMCE HTML field editor**

The following properties are included with the TinyMCE HTML field editor.

**Table 204: Properties installed with the TinyMCE HTML field editor**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.ui.html.toolbar | Configures the editing toolbar for the htmlArea (legacy) editor.  
  - **Type:** string  
  - **Default value:** `formatblock, fontname, fontsize, bold, italic, underline, justifyleft, justifycenter, justifyright, justifyfull, insertorderedlist, insertunorderedlist, outdent, indent, forecolor, hilitecolor`  
  - **Location:** System Properties > UI Properties  
  - **Label:** Configures the editing toolbar for HTML fields |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.ui.html.editor | Determines which HTML field editor to use: TinyMCE (default) or htmlArea (legacy).  
- **Type:** choice list  
- **Default value:** tinymce  
- **Location:** System Properties > UI Properties  
- **Label:** HTML field editor to use |
| glide.ui.html.editor.toolbar.line1 | Configures the first line of the TinyMCE version 2 editing toolbar when the TinyMCE editor is enabled.  
- **Type:** string  
- **Default value:** newdocument,|,bold,italic,underline,strike,|,justifyleft,justifycenter,justifyright,justifyfull,formatselect,fontselect,fontcolor,leftcolor,rightcolor,|,underline,strike,|,strikethrough,|,alignleft,aligncenter,alignright,center,justify,|,indent,|,outdent,|,bullets,numbers,|,numberedlist,|,unorderedlist,|,unorderednumberedlist,|,undo,redo,|,link,unlink,|,cleanup,|,code,|,forecolor,|,backcolor,|,removeformat,|,hr,|,visualblocks,|,sub,|,sup,|,charmap,|,image,|,media,|,spellchecker,|,preview,|,fullscreen |
| glide.ui.html.editor.toolbar.line2 | Configures the second line of the TinyMCE version 2 editing toolbar when the TinyMCE editor is enabled.  
- **Type:** string  
- **Default value:** cut,copy,paste,pastetext,|,search,replace,|,bullist,numlist,|,outdent,indent,blockquote,|,undo,redo,|,link,unlink,cleanup,code,|,forecolor,|,backcolor,|,removeformat,|,hr,|,visualaid,|,sub,sup,|,charmap,image,media,|,spellchecker,|,preview,|,fullscreen |
| glide.ui.html.editor.toolbar.valid.buttons | Contains a list of supported buttons for HTML fields for the TinyMCE version 2 editor.  
- **Type:** string  
- **Default value:** newdocument,bold,italic,underline,strike,|,justifyleft,justifycenter,justifyright,justifyfull,formatselect,fontselect,fontcolor,leftcolor,rightcolor,|,underline,strike,|,strikethrough,|,alignleft,aligncenter,alignright,center,justify,|,indent,|,outdent,|,bullets,numbers,|,numberedlist,|,unorderedlist,|,unorderednumberedlist,|,undo,redo,|,link,unlink,|,cleanup,|,code,|,forecolor,|,backcolor,|,removeformat,|,hr,|,visualblocks,|,sub,|,sup,|,charmap,|,image,|,media,|,spellchecker,|,preview,|,fullscreen |
| glide.ui.html.editor.v4.toolbar.valid.buttons | Contains a list of supported buttons for HTML fields for the TinyMCE version 4 editor.  
- **Type:** string  
- **Default value:** newdocument,bold,italic,underline,strike,|,justifyleft,justifycenter,justifyright,justifyfull,formatselect,fontselect,fontcolor,leftcolor,rightcolor,|,underline,strike,|,strikethrough,|,alignleft,aligncenter,alignright,alignjustify,formatselect,fontselect,fontcolor,leftcolor,rightcolor,|,underline,strike,|,strikethrough,|,alignment,|,indent,|,outdent,|,bullets,numbers,|,numberedlist,|,unorderedlist,|,unorderednumberedlist,|,undo,redo,|,link,unlink,cleanup,code,|,forecolor,|,backcolor,|,removeformat,|,hr,|,visualblocks,|,sub,|,sup,|,charmap,|,image,|,media,|,spellchecker,|,preview,|,fullscreen |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.ui.html.editor.v4.toolbar.line1 | Configures the first line of the TinyMCE version 4 editing toolbar when the TinyMCE editor is enabled.  
  - **Type**: string  
  - **Default value**: bold,italic,underline,undo,redo,|,fontselect,fontsizeselect,tab  
  - **Location**: System Properties > UI Properties  
  - **Label**: (TinyMCE v.4.0.26) Configures the editing toolbar (first line) for HTML fields |
| glide.ui.html.editor.v4.toolbar.line2 | Configures the second line of the TinyMCE version 4 editing toolbar when the TinyMCE editor is enabled.  
  - **Type**: string  
  - **Default value**: none  
  - **Location**: System Properties > UI Properties  
  - **Label**: (TinyMCE v.4.0.26) Configures the editing toolbar (second line) for HTML fields |
| glide.ui.html.editor.v4.font.collection | Contains a list of available font collections for the TinyMCE version 4 editor.  
  - **Type**: string  
  - **Default value**: Andale Mono=andale mono,times;Arial=arial,helvetica,sans-serif;Arial Black=arial black,avant garde;Book Antiqua=book antiqua,palatino;Comic Sans MS=comic sans ms,sans-serif;Courier New=courier new,courier;Georgia=georgia,palatino;Helvetica=helvetica;Impact=impact,chicago;Symbol=symbol;Tahoma=tahoma,arial,helvetica,sans-serif;Terminal=terminal,monaco;Times New Roman=times new roman,times;Trebuchet MS=trebuchet ms,geneva;Verdana=verdana,geneva;Webdings=webdings;Zapf Dingbats;  
  - **Location**: System Properties > UI Properties  
  - **Label**: (TinyMCE v.4.0.26) User can select different font collection for Tiny MCE HTML editor |

Business rules installed with the TinyMCE HTML field editor  
The following business rules are included with the TinyMCE HTML field editor.
Table 205: Business rules installed with the TinyMCE HTML field editor

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>validate_tinymce_buttons</td>
<td>Ensures that the TinyMCE HTML version 3 editor toolbar is configured with valid buttons.</td>
</tr>
<tr>
<td>validate_tinymce4_buttons</td>
<td>Ensures that the TinyMCE HTML version 4 editor toolbar is configured with valid buttons.</td>
</tr>
</tbody>
</table>

Modify image uploads
By default, users can use and upload images to the image library from an HTML field. You can disable access to the image library from HTML fields.

Role required: admin
1. Navigate to System UI > UI Pages.
2. Select html_insert_image_dialog.
3. Locate the following lines of code in the HTML field and comment them out.

```html
<j:if test= "${jvar_use_dbimage}">
  <j:set var = "jvar_default_insert_image_type" value = "dbimage" />
  <g:ui_select_option text = "${gs.getMessage('Image Library')}" value = "dbimage" selected = "${jvar_default_insert_image_type}" />
</j:if>
```
4. Click Update.

To remove the Upload from URL option in the HTML editor, add a new property named glide.ui.html.image.allow_url and set the Value to false.

Add an image field
Image fields enable you to add images to forms.

Role required: personalize_form
For example, you can add portraits to the user records in your system. The image type must be .gif, .jpg/.jpeg, or .png.

Image resizing is defined by the CSS. Larger images are resized to 250 pixels. For most browsers, the larger of the height and width measurements is reduced to 250 pixels and the proportion of the image is maintained. For example, an image with a size of 1508 x 663 pixels is resized to 250 x 110 pixels. Some browsers resize both the height and width measurements to 250 pixels, resulting in a square image.

You can add a new image field.
1. Open the desired form.
2. Create a new field with the Type set to Image.
   For instructions, see Add a new field to a table on page 783.
3. Add the new field to the form and save your customization.
   The form displays the new blank image field.
4. Click Click to add in the image field, select an image to upload, and click OK.
   The selected image is attached to the form and displayed in the image field.
Journal fields

There are three types of journal field: journal, journal_list, and journal_input. Each type of journal field behaves differently.

<table>
<thead>
<tr>
<th>Journal field types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>journal</td>
<td>Allow and store input, and display the combined inputs below the input box.</td>
</tr>
<tr>
<td>journal_input</td>
<td>Allow and store input, but do not display the combined inputs.</td>
</tr>
<tr>
<td>journal_list</td>
<td>Do not allow or store input; they merely display the contents of other Journal fields upon which the journal_list field is dependent. If a journal_list field is dependent on more than one Journal field, it will chronologically interweave those fields’ inputs.</td>
</tr>
</tbody>
</table>

Render journal field entries as HTML

Journal fields can render text enclosed within code tags as HTML.

- Role required: any role that grants write access to a journal field
- System property: the High Security Setting glide.ui.security.allow_codetag is set to the default value of true

By default, a High Security Setting *escapes* any HTML code you type in a journal field by replacing it with its equivalent HTML entity value. Escaping causes the system to display HTML code as text rather than forwarding it to the browser as rendering instructions.

1. Enter \[code\] [/code] tags around any code you want to render as HTML.

**Note:** A single journal entry can contain multiple code tags as long as each code tag has a beginning and ending tag.
For example, enter these lines:

### Table 207: Results of entering HTML code

<table>
<thead>
<tr>
<th>Code entered</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;a href=&quot;http://www.servicenow.com&quot;&gt;ServiceNow&lt;/a&gt;</code></td>
<td>The system renders a hyperlink to the ServiceNow web site.</td>
</tr>
<tr>
<td><code>&lt;b&gt;This text will be bold.&lt;/b&gt;</code></td>
<td>The system renders the sentence in bold.</td>
</tr>
<tr>
<td><code>&lt;b&gt;This text will not be bold.&lt;/b&gt;</code></td>
<td>The system escapes the bold tags and renders them as text.</td>
</tr>
<tr>
<td><code>&lt;script&gt;gs.info(gs.getUserDisplayName());&lt;/script&gt;</code></td>
<td>The system escapes the content of the script tag.</td>
</tr>
</tbody>
</table>

**Note:** By default, the HTML Sanitizer prevents the entry of `<script>` elements.

For more examples of HTML formatting options, see the blog post [Formatting within Journal fields using HTML & [code]](https://community.servicenow.com/b/ServiceNow-TechSupportBlogPosts) by a ServiceNow Technical Support Engineer in the ServiceNow Community.

2. Click **Post**.

**Note:** You cannot edit previous journal entries.

The system renders the text within code tags as HTML.

### Restrict the CODE tag in journal fields

You can prevent journal fields from rendering HTML code by disabling support for the `[code]` tag.

Role required: admin

1. Navigate to **System Properties > UI Properties**.
2. Clear the check box for **Allow support for embedding HTML code by using the [code] tag** (the glide.ui.security.allow_codetag property).
3. Click **Save**.

### Validate HTML in journal fields

You can prevent users from saving invalid HTML in a journal field.

Role required: admin

1. Add the property glide.ui.allow_deep_html_validation. For instructions, see [Add a property using sys_properties.list](https://community.servicenow.com/b/ServiceNow-TechSupportBlogPosts) on page 1411.
2. Set the **Value** to **true** (it is false by default).
3. Click **Save**.

Users now see a warning in the activity formatter when they enter invalid HTML code in a journal field.
System property for restricting the number of entries sent in a notification

Administrators can control the number of journal entries notifications include with the following system property.

<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.email.journal.lines | Number of journal entries (Additional comments, Work notes, etc.) included in email notifications (-1 means all). | Specifies the number of entries from a journal field (such as Additional comments and Work notes) included in email notifications. A value of -1 includes all journal entries.  
  * Type: integer  
  * Default value: 3  
  * Location: System Properties > Email |

Code for getting the contents of a journal field into an array

To put the contents of a journal field into an array so that you can iterate through each entry, you can use the code in this page.

```javascript
var notes = current.work_notes.getJournalEntry(-1);  //gets all journal entries as a string where each entry is delimited by '\n 
var na = notes.split("\n\n");  //stores each entry into an array of strings
for (var i = 0; i < na.length; i++)
gs.print(na[i]);
```

Journal field display limits

Journal fields can greatly increase the size of task records because they allow users to enter very large string values.

Display limits prevent the instance from loading the entire journal field into memory. Administrators have the option to:

- Set the length at which journal fields stop displaying the entire field's contents and instead only display a portion (called a preview) of the field's contents. Users can still access the field's entire contents by clicking a Show All button.
- Set the size of the preview text the journal field displays.
• Set the maximum number of journal entries journal fields can display.

Set the maximum display size for journal fields
You can set the maximum display size for journal fields by adding a system property.

Role required: admin

When a journal field exceeds the length set in this property, the instance shows a preview of the journal field instead of the field's entire contents. The preview includes a Show All button to display the rest of the field's contents. A separate property determines the number of characters the preview displays.

1. Add a system property with the following settings.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.max_journal_list_size</td>
</tr>
<tr>
<td>Description</td>
<td>Size in megabytes when a journal field should display a preview rather than the field's entire contents.</td>
</tr>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Value</td>
<td>10</td>
</tr>
</tbody>
</table>

2. Click Submit.

Set the journal preview size
When a journal field exceeds the size of the glide.max_journal_list_size property, the instance displays a preview rather than the field's entire contents.

Role required: admin

To specify the amount of text to display as a preview, set the following system property. Users can click the Show All button to see the rest of the field's contents.

Add a system property with the following settings. For instructions, see Add a property using sys_properties.list on page 1411.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.shortened_journal_length</td>
</tr>
<tr>
<td>Description</td>
<td>Number of characters to display as a preview of journal fields.</td>
</tr>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Value</td>
<td>512000</td>
</tr>
</tbody>
</table>

Set the maximum number of journal entries
To set the maximum number of entries the system shows in the activity formatter, edit the following system property. The activity formatter displays the entries starting with the most recent entry up to the maximum number.

Role required: admin

1. Enter sys_properties.list in the navigation filter.
2. Search for the property glide.history.max_entries.
3. Edit the Value of the property.
### Journal field script values

The `setValue()` method is not supported for journal fields. Instead, assign values in script as in the following example.

```javascript
var gr = new GlideRecord('incident');
//query priority 1 incidents in the state of either 'new' or 'active'.
gr.addQuery('priority', 1);
var gc = gr.addQuery('state', 1);
gc.addOrCondition('state', 2);
gr.query();
while(gr.next()) {
    //print a list of the incident numbers updated
    gs.print(gr.number);
    //add an entry to the 'work notes' journal field for each incident
    gr.work_notes = "This is a high-priority incident. Please prioritize.";
    gr.update();
}
```

### Percent complete fields

Administrators can create percent complete fields, which accept decimal input and appear as progress bars when displayed in lists.

For example, use a percent complete field to set the completion percentage for a task or project in a form and then see that percentage displayed as a progress bar in a list.

Administrators can also configure views that compare actual progress with a target value to determine if goals are being met and then apply color to provide visual alerts where progress does not meet expectations.

![Percent complete field](image)

**Figure 255: Form view of a percent complete field**
Create a percent complete field

You can add a percent complete field to a form.

Role required: personalize_form

1. Navigate to any form.
2. Right-click the form header and select Configure > Form Layout.
3. Create a new field and set the Type to Percent Complete.
4. Click Add.
   The new field appears at the bottom of the Selected list.
5. Use the up and down arrow buttons to move the field to the desired location in the form.
6. Click Save.
   The form reloads and displays the new field.

Convert a field to a percent complete field

You can convert any existing decimal field to a percent complete field.

Role required: personalize_form

1. Right-click the decimal field's label and select Configure Dictionary.
2. Change the Type field from Decimal to Percent Complete.
3. Click Update.

Add a target field attribute

Add an optional attribute (target_field) to a percent complete field to compare the actual completion percentage of a task or project with a target percentage in a different decimal field that specifies where the task should be at this point.

Role required: personalize_dictionary

If a target field is not specified, the target of 100 is assumed.

1. Right-click the % Complete field in a form.
2. Select Configure Dictionary from the pop-up menu.
3. In the Dictionary Entry form, add the following attribute:
   target_field=percent_complete_target
4. Update the dictionary record.
   In the list, a gray bar appears behind the colored bar to indicate the target value. The gray target bar appears only if you defined a target field.
Target threshold colors attribute

If the `target_field` attribute is configured, a second attribute called `target_threshold_colors` enables an administrator to define additional parameters.

The parameters are:

- Different thresholds at which the colored bar should change color
- A specific color for each threshold

The format of this attribute's value is `number1:color1;number2:color2` and so on. Use this attribute to apply warning colors to completion percentages that are lower than target percentages. These values are defined as the percentage of target accomplished. For example, a value of `0:red;50:yellow;90:green` displays a red bar if the progress to target percentage is between 0-49. If the percent of target is between 50 and 89, the color is yellow. Percent of target 90 and above displays in green. Completion percentages that exceed target percentages also display in green. Order the color attributes from the smallest percentage to the largest.

If you do not specify a `target_field`, then a target of 100 is used, allowing you to use the color thresholds with a single field value.

```plaintext
target_field=percent_complete_target,target_threshold_colors=0:tomato;50:khaki;90:lightgreen
```

The following table lists examples of percent of target calculation using the colors defined above.

<table>
<thead>
<tr>
<th>Target percent</th>
<th>Percent field value</th>
<th>Percent of target calculation</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>40</td>
<td>40%</td>
<td>tomato</td>
</tr>
<tr>
<td>65</td>
<td>59</td>
<td>90.7%</td>
<td>lightgreen</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>66.7%</td>
<td>khaki</td>
</tr>
</tbody>
</table>
Phone number fields

The E.164 phone number standard ensures that all necessary information for a phone number is included and properly formatted to successfully route an international call over a territory's public telephone network.

When a user enters a phone number, it is received and stored as a string of numbers. An E.164 phone number field automatically formats and validates the numbers so that they are E.164-compliant when displayed as local and international numbers. The E.164 phone number field type does not replace the phone field type.

E.164 phone number field characteristics

What an E.164 phone number field displays.

An E.164 phone number field displays:

- [Optional] A choice list for the phone number territory.
- [Always] An input box for entering phone numbers.
- [By Default] A red underline when a phone number does not match the format for the selected phone territory and cannot be saved.
- [Optional] A green underline when a phone number does not match the format for the selected phone territory but can be saved with Other / Unknown as the territory.

Territories assigned

 Territories are assigned to locations, and are not assigned directly to users.

A user's territory, and so the user's E.164-compliant phone functionality, is based on the user's location. For example, if a user has a location of **SHS quadra 5, Bloco E., Brasilia** defined in the User [sys_user] table, the parent record for Brazil in the location table defines the phone territory. The phone territory may be assigned at any level of the **locations** hierarchy, which is searched going up to the next parent until the territory is found or no parents remain.
Dependent fields

In the dictionary, you can specify a dependent field in the User or Location field, which displays the appropriate territory in the selector choice list when a user enters a phone number.

In the dictionary, you can specify a dependent field in the User or Location field, which displays the appropriate territory in the selector choice list when a user enters a phone number. For example, if you enter caller_id in the dependent field in the Incident table, the appropriate territory is added to the territory selector choice list when a user enters caller information.

E.164 phone number field configuration

Administrators can use the phone number system properties and dictionary attributes to do certain things. The system properties apply the configuration option to all phone number fields that do not have a comparable dictionary attribute. The dictionary attributes apply the configuration only to the phone number field it is added to. Since dictionary attributes take precedence over system properties, administrators can set a global configuration with a property and then apply exceptions on a field-by-field basis.

Requiring territory format validation

By default, phone number fields require that a phone number match the display format of the selected territory.

By default, phone number fields require that a phone number match the display format of the selected territory. If a phone number does not match this format, the input box displays a red line underneath the phone number and users are prevented from saving it.

![Call Back Number: North America +185585551212](image)

Figure 259: Phone e164 strict

Setting the glide.phone_number_e164.strict system property to false or adding the pn_strict dictionary attribute allows the phone number input box to display a green line underneath numbers that do not match the territory format listed for the selected territory. In this case, a user can save a phone number in an invalid format, but the field continues to display a warning until the phone number matches the format required by the territory. You can use the Other / Unknown territory to store otherwise invalid phone numbers.

![Call Back Number: North America +185585551212](image)

Figure 260: Phone e164 not strict

⚠️ Warning: Switching from optional territory format validation to required territory format validation may result in some phone numbers failing validation altogether. In such cases, the E.164 phone number field displays an error message.

Requiring entry of international format

By default, users can enter phone numbers in their territory's local format and do not have to format the number for international dialing.

The phone number field automatically formats local phone numbers into E.164-compliant international phone numbers when the user finishes editing the field. As long as the phone number entered matches the territory's format for a local number, users can save the phone number.
Changing the `glide.phone_number_e164.allow_national_entry` system property or adding the `pn_allow_national_entry` dictionary attribute requires users to enter a phone number in the territory's international format, which starts with the plus (+) character. Users cannot save a locally formatted phone number, and the phone number input box displays a red line underneath phone numbers without the proper international formatting.

**Configure the display of the local format**

By default, an E.164 phone number field always displays phone numbers in an international format.

Change the E.164 phone number to use the local format in certain circumstances by adding the following system property:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>glide.phone_number_e164.display_national</code></td>
<td></td>
</tr>
</tbody>
</table>
|  • **Type**: string  
  • **Default value**: false  
  • **Other possible values**:  
    • **true or form**: displays phone numbers in a local format on forms, but displays an international format on lists.  
    • **all**: always displays phone numbers in a local format.  
    • **user**: only displays phone numbers in a local format when the phone number matches the local setting of the current user.  
    • **false**: does not display phone numbers in local format. |

Select one of the possible values to determine how the system handles the E.164 phone number.

You can also - to a specific field to override the system property:

<table>
<thead>
<tr>
<th>Dictionary attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pn_display_national</code></td>
<td>Overrides the <code>glide.phone_number_e164.display_national</code> property setting for how an E.164 phone number is displayed.</td>
</tr>
</tbody>
</table>
Configure the phone territory selector choice list
By default, an E.164 phone number field always displays the phone territory associated with the phone number.

Changing the system property or adding the dictionary attribute hides the territory selector choice list. If the territory selector choice list is hidden, users can only enter a local or national number.

Figure 264: Phone e164 hide selector

Add the following system property to show or hide the territory selector choice list:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.phone_number_e164.display_territory_selector</td>
<td>Determines whether to display the territory selector choice list. Hiding the territory selector choice list restricts users to entering only local or national phone numbers.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: true</td>
</tr>
</tbody>
</table>

You can also add the following dictionary attribute to a specific field to override the system property:

<table>
<thead>
<tr>
<th>Dictionary attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pn_display_territory_selector</td>
<td>Overrides the glide.phone_number_e164.display_territory_selector property setting that determines whether to display the territory selector choice list. Available values are identical to those described above for the glide.phone_number_e164.display_territory_selector property.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Example</strong>: pn_display_territory_selector=false</td>
</tr>
</tbody>
</table>

Configure the display of territory labels
A property controls how territory labels are displayed.
You can display territory labels to the right of the number in an E.164 phone number field by setting the system property or adding the dictionary attribute. This is useful if the territory selector choice list is turned off and you want the user to see the territory for the entered phone number.

Figure 265: E164 phone display territory labels

Enabling territory labels also displays the phone territory in lists.

*Add the following system property* to display the territory label to the right of the number in an E.164 phone number.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.phone_number_e164.display_territory_text</td>
<td>Determines when an E.164 phone number field displays a territory label.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: string</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: read-only</td>
</tr>
<tr>
<td></td>
<td>• <strong>Other possible values</strong>:</td>
</tr>
<tr>
<td></td>
<td>• <strong>all</strong>: always displays the territory label.</td>
</tr>
<tr>
<td></td>
<td>• <strong>national</strong>: displays the territory label only if the phone number is in local format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>read-only</strong>: displays the territory label in read-only mode, regardless of whether the number is in local or international format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>read-only-national</strong>: displays the territory label in read-only mode only if the number is in local format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>list</strong>: displays the territory label in a list.</td>
</tr>
<tr>
<td></td>
<td>• <strong>list-national</strong>: displays the territory label in a list if the number is in national format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>none</strong>: does not display the territory label.</td>
</tr>
</tbody>
</table>

You can also *add the following dictionary attribute* to a specific field to override the system property:

<table>
<thead>
<tr>
<th>Dictionary attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pn_display_territory_text</td>
<td>Overrides the glide.phone_number_e164.display_territory_text property that defines when a phone number field displays a territory label. Available values are identical to those described above for the glide.phone_number_e164.display_territory_text property.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: string</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: read-only</td>
</tr>
<tr>
<td></td>
<td>• <strong>Example</strong>: pn_display_territory_text=all</td>
</tr>
</tbody>
</table>
Configure the international direct dialing prefixes
A property is available to control the display of prefixes.

You can enable the display of the international direct dialing prefix, which appears between the territory selector choice list and the input box for an E.164 phone number field on forms, by setting the system property or adding the dictionary attribute.

Figure 266: E164 phone display idd

Add the following system property to display the international direct dialing prefix.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.phone_number_e164.display_users_idd</td>
<td>Determines whether to display the international direct dialing prefix between the territory selector choice list and the input box on forms.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
</tbody>
</table>

You can also add the following dictionary attribute to a specific field to override the system property:

<table>
<thead>
<tr>
<th>Dictionary attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pn_display_users_idd</td>
<td>Overrides the glide.phone_number_e164.display_users_idd property that determines whether to display the international direct dialing prefix between the territory selector choice list and the input box on forms. Available values are identical to those described above for the glide.phone_number_e164.display_users_idd property.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Example: pn_display_users_idd=false</td>
</tr>
</tbody>
</table>

E.164 phone number field system properties

These properties are available to configure E.164 phone number fields.

**Note:** To open the System Properties [sys_properties] table, enter `sys_properties.list` in the navigation filter.
### Table 209: e.164 phone number field system properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.phone_number_e164.strict</td>
<td>Determines whether all phone number fields must match the display format for the selected territory.</td>
</tr>
<tr>
<td></td>
<td>When the value is <strong>true</strong>, the phone number input box displays a red line underneath phone numbers that do not match the format for the selected territory. Users cannot save the phone number.</td>
</tr>
<tr>
<td></td>
<td>When the value is <strong>false</strong>, the phone number input box displays a green line underneath phone numbers that do not match the format for the selected territory. Users can save the phone number. The territory selector choice list offers the option to select an <strong>Other / Unknown</strong> territory format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: Add to the System Properties [sys_properties] table.</td>
</tr>
<tr>
<td>glide.phone_number_e164.allow_national_entry</td>
<td>Determines whether users can enter phone numbers in the local format or whether they must enter phone numbers in international format.</td>
</tr>
<tr>
<td></td>
<td>When <strong>true</strong>, users can enter phone numbers in the local format for the selected territory. When <strong>false</strong>, users must enter phone numbers in the international format for the selected territory.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: Add to the System Properties [sys_properties] table.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.phone_number_e164.display_national</td>
<td>Determines whether to display E.164 phone numbers in local format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: string</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: false</td>
</tr>
<tr>
<td></td>
<td>• <strong>Other possible values:</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>true or form</strong>: displays phone numbers in a local format on forms, but displays an international format on lists.</td>
</tr>
<tr>
<td></td>
<td>• <strong>all</strong>: always displays phone numbers in a local format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>user</strong>: only displays phone numbers in a local format when the phone number matches the local setting of the current user.</td>
</tr>
<tr>
<td></td>
<td>• <strong>false</strong>: does not display phone numbers in local format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: Add to the System Properties [sys_properties] table.</td>
</tr>
<tr>
<td>glide.phone_number_e164.display_territory_selector</td>
<td>Determines whether to display the territory selector choice list.</td>
</tr>
<tr>
<td></td>
<td>Hiding the territory selector choice list restricts users to entering only local or national phone numbers.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: Add to the System Properties [sys_properties] table.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.phone_number_e164.display_territory_text</td>
<td>Determines when an E.164 phone number field displays a territory label.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> string</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> read-only</td>
</tr>
<tr>
<td></td>
<td>• <strong>Other possible values:</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>all:</strong> always displays the territory label.</td>
</tr>
<tr>
<td></td>
<td>• <strong>national:</strong> displays the territory label only if the phone number is in local format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>read-only:</strong> displays the territory label in read-only mode, regardless of whether the number is in local or international format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>read-only-national:</strong> displays the territory label in read-only mode only if the number is in local format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>list:</strong> displays the territory label in a list.</td>
</tr>
<tr>
<td></td>
<td>• <strong>list-national:</strong> displays the territory label in a list if the number is in national format.</td>
</tr>
<tr>
<td></td>
<td>• <strong>none:</strong> does not display the territory label.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location:</strong> Add to the System Properties [sys_properties] table.</td>
</tr>
</tbody>
</table>

| glide.phone_number_e164.display_users_idd    | Determines whether to display the international direct dialing prefix between the territory selector choice list and the input box on forms.                                                               |
|                                              | • **Type:** true | false                                                                                                                                  |
|                                              | • **Default value:** false                                                                                                                        |
|                                              | • **Location:** Add to the System Properties [sys_properties] table.                                                                            |

**E.164 phone number field dictionary attributes**

You can override the global system property with certain dictionary attributes.

**Table 210: e.164 phone number field dictionary attributes**

<table>
<thead>
<tr>
<th>Dictionary attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pn_strict</td>
<td>Overrides the glide.phone_number_e164.strict property setting that requires all phone number entries match the format for the selected territory. Available values are identical to those described above for the glide.phone_number_e164.strict property.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Example:</strong> pn_strict=false</td>
</tr>
<tr>
<td>Dictionary attribute</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pn_allow_national_entry</td>
<td>Overrides the glide.phone_number_e164.allow_national_entry property setting that determines whether users can enter phone numbers in the local format or whether they must enter phone numbers in international format. Available values are identical to those described above for the glide.phone_number_e164.allow_national_entry property.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Example: pn.allow_national_entry=false</td>
</tr>
<tr>
<td>pn_display_national</td>
<td>Overrides the glide.phone_number_e164.display_national property setting for how an E.164 phone number field displays phone numbers. Available values are identical to those described above for the glide.phone_number_e164.display_national property.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Example: pn.display_national=all</td>
</tr>
<tr>
<td>pn_display_territory_selector</td>
<td>Overrides the glide.phone_number_e164.display_territory_selector property setting that determines whether to display the territory selector choice list. Available values are identical to those described above for the glide.phone_number_e164.display_territory_selector property.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Example: pn.display_territory_selector=false</td>
</tr>
<tr>
<td>pn_display_territory_text</td>
<td>Overrides the glide.phone_number_e164.display_territory_text property that defines when a phone number field displays a territory label. Available values are identical to those described above for the glide.phone_number_e164.display_territory_text property.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: read-only</td>
</tr>
<tr>
<td></td>
<td>• Example: pn.display_territory_text=all</td>
</tr>
</tbody>
</table>
### Dictionary attribute

<table>
<thead>
<tr>
<th>Dictionary attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pn_display_users_idd</td>
<td>Overrides the glide.phone_number_e164.display_users_idd property that determines whether to display the international direct dialing prefix between the territory selector choice list and the input box on forms. Available values are identical to those described above for the glide.phone_number_e164.display_users_idd property.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Example: pn_display_users_idd=false</td>
</tr>
</tbody>
</table>

**Configure a territory phone display rule**

The string of numbers that make up a phone number is automatically validated and formatted for a specific territory by applying a series of regular expressions.

**Role required: admin**

The number is first validated against the phone validations that have been defined for the territory, and in the order specified by the Order field. To be valid, the number must match the regular expression defined in the Condition field for at least one phone validation.

After a number has been validated, the Condition expression for each format defined for the territory is applied to the number in the order determined by the Order fields. The Pattern and Format regular expressions are applied to produce a phone number that is formatted correctly for the territory.

The Sys Phone Territory screen allows administrators to edit the display rules for a given territory. Administrators may want to modify the Active, Display, or Order fields. To edit the display rules for a territory:

2. Click a territory Name.
3. Edit the fields, as appropriate (see table).
4. Click Update.
### Table 211: Territory phone display rules

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The unique name of the territory.</td>
</tr>
<tr>
<td>Country calling code</td>
<td>The <em>country code</em> for dialing numbers from outside the territory.</td>
</tr>
<tr>
<td>International direct dial</td>
<td>The prefix for calling internationally from the territory, such as 00 or 001.</td>
</tr>
<tr>
<td>STD</td>
<td>The subscriber trunk dialing code, also known as the direct distance dialing code, which is a sequence of numbers before the telephone number that indicate whether the call is to be routed outside of the local calling area.</td>
</tr>
<tr>
<td>International prefix</td>
<td>The prefix required to dial an international call, such as a plus sign (+).</td>
</tr>
<tr>
<td>National prefix</td>
<td>The prefix required to dial a local call.</td>
</tr>
<tr>
<td>Active</td>
<td>An indicator for whether the territory phone definition is active. If deactivated, this territory unavailable to users.</td>
</tr>
<tr>
<td>Trunk dialing code optional</td>
<td>An indicator for whether the STD code is optional.</td>
</tr>
<tr>
<td>STD follows country</td>
<td>An indicator for whether the STD code should be displayed to the right of the country calling code.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Display</td>
<td>An indicator for whether to display the territory in the choice list. Clearing this check box removes the territory from the choice list. If an international number is entered for a territory that is not displayed in the territory selector choice list, that territory is temporarily added to the selector choice list for that field only. For example, if the United Kingdom Display field is not selected, the United Kingdom does not appear in the territory selector choice list. However, if the user enters an international number beginning with +44, the United Kingdom is added to the list and the number is formatted and validated accordingly.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which a territory appears in a choice list. Territories are sorted numerically by the number assigned here. If more than one territory is assigned the same number, they are subsorted alphabetically. All territories are assigned a default value of 100. To display a territory at the top of the list, assign a value that is less than 100. To display a territory at the end of the list, assign a value that is greater than 100. For example, if a territory is assigned an order of 500, it is displayed at the end of the list, and if more than one territory is assigned an order of 500, they are listed alphabetically at the end of the list.</td>
</tr>
</tbody>
</table>

**Phone validations**

Phone validations are already configured for all territories and are automatically applied to the phone number to ensure that the number is valid for the territory.
Figure 267: E164 phone validations

**Phone formats**
Phone formats are already configured for all territories and are automatically applied to the phone number to ensure that the number is valid for the territory.

Figure 268: E164 phone formats

**Watch lists**
Watch lists allow multiple users to subscribe to notifications of a task.

Users must have an email address defined in their user record or enter an email address into the watch list email field to receive watch list notifications.

**Important:** Administrators configure email notifications for watch lists (see Watch list configuration on page 838).
Use a watch list

Using a watch list includes expanding the watch list and selecting users with the glide controls.

Role required: none

1. Expand the watch list by clicking the lock icon (🔒/🔓).
2. Select users with the glide list controls (see table).

<table>
<thead>
<tr>
<th>UI16/UI15 icon</th>
<th>UI11 icon</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>🔄</td>
<td>Add the current user</td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>Remove the highlighted user</td>
</tr>
<tr>
<td>🛡</td>
<td>🛡</td>
<td>Open the highlighted user's record (active only when a user record exists)</td>
</tr>
<tr>
<td>🕵</td>
<td>🕵</td>
<td>Open a slushbucket to add or remove multiple users (not available until a record has initially been saved)</td>
</tr>
<tr>
<td>🗑</td>
<td>🗑</td>
<td>Collapse or expand the watch list</td>
</tr>
</tbody>
</table>
### Watch list configuration

Watch lists (glide_list field type) allow multiple users to subscribe to notifications of a task.

Administrators can configure options for watch list fields.

**Configure email notifications for watch lists**

You can specify in an email notification that members of the watch list receive the notification when the conditions are met.

Role required: admin

To receive these notifications, users must define an email address in their user record or enter an email address into the watch list email field.

1. Open the notification to configure.
2. In the Who will receive section, select the icon beside Users/groups in fields.
3. Double-click Watch list in the Available column to move it to the Selected column.
4. Click Update.

**Multiple watch lists**

An advanced configuration using watch lists might involve placing two watch lists on a form, one for the general comments on a task and another for work notes or non-public comments. By configuring separate email notifications, separate users on each watch list can be notified about different information.

**Troubleshooting duplicate email notifications**

If users on a watch lists are getting more than one email for each update to an incident, it may be because the people involved in the incident are replying to an email notification in which other user's names appear in the To: or CC: field. Recipients may be receiving email through their email system (Outlook, Groupwise, etc.) and through ServiceNow ITSA Suite. To stop this duplication, remove the names of other users from the email or the watch list.

**Hide email addresses in a watch list**

You can remove the email address text entry element from a watch list by modifying the dictionary.

Role required: admin

1. Open a task record that displays the Watch list field.
2. Right-click the label and select Configure Dictionary.
3. In the Attributes related list, click New.
4. Enter or select the no_email attribute and enter true in the Value field.
5. Click Submit.

The email entry field is hidden. Users can select users from the reference field to add to the watchlist.
Wiki fields

Wiki text fields use the basic wiki markup language and support links to external URLs and to sources within the system.

Some fields in the base system support wiki text and administrators can add wiki text fields on any form in the system.

Supported wiki tags

Basic tags are supported for wiki text fields in the system.

<table>
<thead>
<tr>
<th>Format</th>
<th>Wiki tag</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Headers</strong></td>
<td>= Header 1 =, == Header 2 ==, etc.</td>
</tr>
<tr>
<td><strong>Numbered step</strong></td>
<td>#Step 1</td>
</tr>
<tr>
<td><strong>Bullets (multi-level)</strong></td>
<td>*Bullet 1, **Bullet 2</td>
</tr>
<tr>
<td><strong>Indentation</strong></td>
<td>::Level 1, ::::Level 2</td>
</tr>
<tr>
<td><strong>Line break</strong></td>
<td>&lt;br&gt;</td>
</tr>
<tr>
<td><strong>Tables</strong></td>
<td>Full table support, including cell spacing, padding, borders, background shading, and width measurements.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Sortable tables are not supported (class=&quot;sortable&quot;).</td>
</tr>
<tr>
<td><strong>Code blocks</strong></td>
<td>&lt;pre&gt; Generic block &lt;/pre&gt;</td>
</tr>
<tr>
<td><strong>Bold</strong></td>
<td>'''text'''</td>
</tr>
<tr>
<td><strong>Italics</strong></td>
<td>''text''</td>
</tr>
<tr>
<td><strong>Ignore Wiki and HTML formatting</strong></td>
<td>&lt;nowiki&gt; wikitext &lt;/nowiki&gt;</td>
</tr>
<tr>
<td><strong>Web link</strong></td>
<td>[<a href="http://community.service-now.com/">http://community.service-now.com/</a> &lt;link text&gt;]</td>
</tr>
<tr>
<td><strong>Placed images</strong></td>
<td>[[Image:&lt;image name&gt;]] See for details on uploading image files to the database.</td>
</tr>
<tr>
<td><strong>HTML</strong></td>
<td>Most common HTML tags are supported</td>
</tr>
<tr>
<td></td>
<td>&lt;b&gt;bold&lt;/b&gt;, &lt;strong&gt;strong&lt;/strong&gt;, &lt;i&gt;italics&lt;/i&gt;, &lt;h1&gt;header1&lt;/h1&gt;, &lt;h2&gt;header2&lt;/h2&gt;, &lt;h3&gt;header3&lt;/h3&gt;, &lt;p&gt;paragraph&lt;/p&gt;, &lt;sub&gt;sub&lt;/sub&gt;, &lt;sup&gt;sup&lt;/sup&gt;, &lt;center&gt;center&lt;/center&gt;</td>
</tr>
</tbody>
</table>

Create a wikitext field

You can create a wikitext field.
Role required: personalize_form

1. Create a new field with the **Type** set to **Wiki**.
2. Click **Wikitext** to begin editing.

The wiki field appears on the form.

**Extend the functionality of a wikitext field**

Configure the wiki field to link to other sources within the system.

Role required: personalize_dictionary

1. Right-click in the header of the new wiki field.
2. Select **Configure Dictionary**
3. In the **Dependent** field of the dictionary form, enter the field you want to use for linking to other pages.
   
   For example, on the Incident table, you might choose the **number** field.
This configuration enables you to link to any incident by using that incident’s number. For example, to open INC0000002, you enter `[[INC0000002]]`.

The following *dictionary attributes* are available for *wiki_text* fields.

- `preview_first=true` - sets the preview mode to display on page load, otherwise editor will be displayed
- `preview_selector=true` - enables the toggle button to switch between the editor and preview
- `dual_mode=true` - enables displaying both the editor and the preview simultaneously during edit mode.

**Choice lists**

A choice list is a type of field that lets the user select from a pre-defined set of choices.

Administrators can define the available choices and customize the behavior and appearance of choice lists.
Choice list security

The personalize_choices security role can be used to let non-administrators modify the options for Choice elements on all tables.

If more granular control is desired, you can also create a custom ACL (security rule) governing the personalize_choices operation either for a particular field or for all fields (\*) on a particular table. However, access to the personalize_choices operation on a particular field does not confer the ability to add new choices for that field. To be able to create new choices for a particular field, two ACLs are required:

- An ACL that grants personalize_choices access for that field.
- An ACL that grants create access to the Choice table (sys_choice).

For example, to give the hris_admin role the ability to personalize only the Category field for Human Resources KB articles, you need:

- An ACL granting personalize_choices access to the hris_admin role on the Category field of the Knowledge (kb_knowledge) table.
- An ACL granting create access to the hris_admin role on the Choice (sys_choice) table.

There are predefined ACLs granting both types of access to the personalize_choices security role, for all fields on all tables. The personalize_choices security role also has read, write, and delete access to the sys_choices table. However, this additional access is not required when making just the Personalize Choices functionality available on a granular basis.

View choice list definitions

You can view the values that appear in choice lists in the Choice [sys_choice] table.

Role required: personalize_choices

The Choice Set [sys_choice_set] table contains a record for every field that uses a choice list. The choice set record is associated with an application file, which allows update sets and team development to track and transfer all choices for a field in a single update record.

Choice list values allow a maximum length of 40 characters.

1. Right-click the choice list field label and select **Show Choice List**. To view other choice list values, modify the filter at the top of the list.
2. Review the items in the list.
Warning: Do not add new choices to the list. To add new choices to a choice list field, use the Configure Choices option.

Define an option for a choice list
How to define a static list of available options.

Role required: personalize_choices

1. Navigate to a form where the field appears.
2. If the choice list is dependent on another field, enter the choice value that the options depend on.
   For example, on the incident table, the Subcategory is dependent on the Category. To customize which subcategory choices are available for the hardware category, select Hardware in the Category field.
3. Right-click the field label and select Configure Choices.
4. Use the slushbucket to rearrange the order, add, or remove items or to create new items.
5. Click Save.

   To dynamically add items to a choice list, use the addOption GlideForm method.

Note: Some business rules may be affected by changes to choice list options (for example, default Incident states).

Reuse a choice list
After defining a set of choice list values, you can reuse the values for another field in a different table.

Role required: personalize_choices

1. Right-click an existing choice field (Field A) and select Configure Choices.
2. Add the desired choice list values.
3. To reuse the choice list values for another field (Field B) in a different table, right-click the label for Field B and select Configure Dictionary.
4. In the Choice table field, select the table where Field A resides.
5. In the Choice field field, select Field A.
6. Click Update.

The choice list values defined on Field A are displayed on Field B. When you add or remove choice list values on Field A, those changes are also reflected on Field B.

Remove the None option
How to remove the None option from a choice list.

Role required: personalize_dictionary

1. Navigate to a form where the field appears.
2. Right-click the field label and select **Configure Dictionary**.
3. Change the Choice field value to **Dropdown without -- None -- (must specify a default value)**.
4. Ensure that the **Default** field is populated to determine which choice is displayed by default.

   **Note:** If the field is dependent on another field, the **-- None --** option remains available.

*Change the None display value*

You can change the default display label of the **None** option for a choice field.

Role required: personalize_choices

1. Navigate to a form on which the field appears.
2. Right-click the field label and select **Show Choice List**.
3. Click **New**.
4. Complete the form.

<table>
<thead>
<tr>
<th>Table 214: Choice form fields</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
</tr>
<tr>
<td>Table</td>
</tr>
<tr>
<td>Element</td>
</tr>
<tr>
<td>Language</td>
</tr>
<tr>
<td>Sequence</td>
</tr>
<tr>
<td>Inactive</td>
</tr>
<tr>
<td>Field</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Label</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Dependent value</td>
</tr>
<tr>
<td>Hint</td>
</tr>
</tbody>
</table>

5. Click **Submit**.

*Delete all choice list options*
You can delete all choices for a choice field from the Choice Set record.

Role required: personalize
You may want to use this method when you are developing a new application and the business requirements change. If you are updating a choice list that is already in use, consider deactivating the options you no longer use to avoid conflicts with existing data or scripts that may rely on the previous options.

1. In the navigation filter, enter `sys_choice_set.list` and press Enter.
2. Open the choice set record for the field.
   
   For example, to locate the choice set for the incident subcategory, filter by `[Table] [is] [incident] AND [Element] [is] [subcategory].
3. Check the box beside the choice set record to delete and select Delete from the Actions choice list below the list.
4. Click Delete in the confirmation window.
   
   All choices for the field are deleted.

Create a choice list for another field type

You can create a choice list for a field with another type (for example, an Integer field).

Role required: personalize_dictionary

You can use this configuration to standardize data entry and limit available options for a field while still maintaining the original field type.

1. Navigate to System Definition > Dictionary.
2. Open the dictionary entry for the field.
3. Change the Choice value to Dropdown with --- None --- or Dropdown without --- None --- (must specify a default value).
4. Right-click the form header and select Save.
5. Click Create Choice List.
   
   • The Choices related list appears on the dictionary entry form.
   • If records on the table contains data for the field, a choice list value for each unique field value is created. For example, if three records exist on the table and each record has a unique value in the field, then three choices are created.
   • If no data exists in the field, a choice list value of -- New choice -- is created.

Display invalid choice list values

By default, inactive or invalid choice list values appear in blue text instead of black. You can disable the color indicator for invalid choices.

Role required: admin

In the following example, the Network category has been deactivated, so it appears in blue for records that still contain the inactive value.
1. Navigate to **System Properties > UI Properties**.
2. Clear the check box for the **Display missing choice list entries** property.

### Adjust the choice list width

By default in UI11, all choice lists use a width of 160 pixels. You can adjust the width for all choice lists on the instance, or change the value for a particular list. This procedure is not applicable to newer versions of the UI.

**Role required:** admin or personalize_styles

1. To change the width for all choice list on the instance, complete the following steps (requires admin role).
   a) Navigate to **System Properties > UI Properties**.
   b) Change the value for the **Default choice list width (pixels)** property (glide.uiinstancedefautwidth).
   c) Click **Save**.

2. To change the value for a particular list, for example, a list with much longer option names than other choice lists, complete the following steps.
   a) Navigate to a form where the field appears.
   b) Right-click the field label and select **Configure Styles**.
   c) Click **New**.
   d) In the **Style** field, enter **width:auto**.
      Leave the **Value** field empty so that the field style applies to all the choices for the field.

   a) Click **Submit**.

      When the field is displayed, the width adjusts to the size of the content.

### Values to associate with choice labels for scripting

When you write a script that references a choice list, you need to know the value that is associated with each choice.
For example, to check whether the incident_state field is active, you could not use the condition `current.incident_state == "active"` because the value associated with the choice labeled **Active** is the integer 2. Instead, you would use the condition `current.incident_state == 2`.

The **Type** field on the choice list dictionary entry determines the data type of the values.

To determine the value associated with a choice, right-click the field label and select **Show Choice List**, and then locate the choice for which you need to know the value.

![Choice List Table](image)

**Figure 271: Show choices list values**

The **-- None --** option may not have a sys_choice record associated with it. A choice list field set to **-- None --** evaluates to these values, depending on the script context as listed below.

- For client-side scripts, such as client scripts: "" (empty string)
- For server-side scripts, such as business rules: "0" (string of the number zero)

**Integer values for default choice lists**

Queries on choice list fields use the value, not the label.

Some common choice lists use integer values that do not match the string labels. For example, the Problem table uses these default values for the **State** field.

<table>
<thead>
<tr>
<th>Value</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open</td>
</tr>
<tr>
<td>2</td>
<td>Known Error</td>
</tr>
<tr>
<td>3</td>
<td>Pending Change</td>
</tr>
<tr>
<td>4</td>
<td>Closed/Resolved</td>
</tr>
</tbody>
</table>
These integer values are also used in several default business rules. For example, a business rule on the Incident table sets the active flag to false when the State field changes to 7, which is the default value for the Closed. If you change the values of your Incident state options, this business rule may no longer behave as desired or expected.

On the Incident table, the Active, State, and Incident state fields are affected by the following default business rules.

### Table 216: Default business rules

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mark_closed (incident)</td>
<td>If the incident_state changes to 7 (Closed), the Active field is set to false</td>
</tr>
<tr>
<td>incident reopen (incident)</td>
<td>If the incident_state is less than 7 (Closed) and the Active field is false, the Active field is set to true</td>
</tr>
<tr>
<td>mark closed (task)</td>
<td>If the state changes to either 3 (Closed Complete) or 4 (Closed Incomplete), the Active field is set to false</td>
</tr>
<tr>
<td>task closer (task)</td>
<td>If the Active flag changes from true to false and the state is neither 3 (Closed Complete) nor 4 (Closed Incomplete), the state is set to 3 (Closed Complete)</td>
</tr>
<tr>
<td>task reopener (task)</td>
<td>If the Active field changes from false to true and the state is either 3 (Closed Complete) or 4 (Closed Incomplete), the state is set to 1 (Open)</td>
</tr>
</tbody>
</table>

**Note:** Notice that these business rules do not change incident_state based on a change to either the Active field or the State field. Changes to incident_state drive the other two fields, not the other way around.

### Best practices for state field choice values

When you configure choice values for the state field, keep a few things in mind.

- Use a negative value to add a new active state field.
- Search for and study the business rules that use a state number filter on the Script and Conditions fields. You can use the Debug tool to trace the order of the business rule execution.
- New values representing inactive states should have a value above 8.

**State modification examples**

Follow these examples for modifying the states of incidents and change requests.

Role required: admin

1. Navigate to System Definition > Choice Lists.
2. At the top of the list, construct a list filter like the following:
   - **Table:** incident
   - **Element:** incident_state
3. Run the filter.
Notice that the Closed state has a value of 7 and the Resolved state has a value of 6. Any state greater than or equal to 7 is assumed to be inactive. Therefore, you should use a positive integer greater than 7 if you want to add a new inactive-type of state. Use a negative value like -1 or -2 if you wish to add a new active-type of state field, such as Awaiting Vendor.

5. At the top of the list, construct a list filter like the following:
   • Table: change_request
   • Element: phase_state

6. Run the filter.
   Notice that the Complete state has a value of 8. Any state greater than or equal to 8 is assumed to be inactive. Therefore, you should use a positive integer greater than 8 if you want to add a new inactive-type of state, such as Cancelled. Use a negative value like -1 or -2 if you wish to add a new active-type of state field, such as Pending.

Troubleshoot change states and business rules
Business rules in the system make assumptions about state values. You can troubleshoot business rules to see the order in which they run and see how it affects changes you make to State field values.

Role required: admin
1. Navigate to System Definition > Business Rules.
2. Construct a filter like this one to view the scripts and conditions that pertain to the Resolved incident_state of 6 or the Closed incident_state value of 7:
   
   The Script field contains 7 OR the Condition field contains 7 OR the Script field contains 6 OR the Condition field contains 6 AND the Table field is incident AND the Active field is true.

See Debug Business Rule for information on how to trace the order of business rule execution. You can click Debug All, resolve an incident, and then check the trace at the bottom of form to watch the business rules execute. These two line examples show that the mark_closed business rule code is entered =>> and then exited <==

```plaintext
=>> 'mark_closed' on incident
<= 'mark_closed' on incident
```

Define field styles
Field styles allow you to declare individual CSS styles for a field in a list or form.

Role required: personalize_styles or admin

The CSS lets you change the following attributes of a field.

- Change the color.
- Change the font attributes (bold, italics, underline).
- Change the padding and alignment of text.

You can define field styles for tables and database views that are in the same scope as the field style and for other tables that have at least one field in the same scope as the field style. Note that field styles defined for a table will not apply to database views that include the table. You must create separate field styles for database views.

1. Navigate to the list of styles for the field in one of the following ways.
• Right-click the field label on the form and select **Configure Styles**.
• Navigate to **System UI > Field Styles**, and locate the field to apply a style (admin only).

2. Click **New**, or click the style to modify.
3. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field name</td>
<td>The field to which the style applies.</td>
</tr>
<tr>
<td>Value</td>
<td>The exact value or script-based-condition required to apply the style.</td>
</tr>
<tr>
<td>Style</td>
<td>The CSS style to apply. For example, to make the background of the Incident Category fields red in record lists, with a 24px font size and white text color, enter the following code.</td>
</tr>
</tbody>
</table>

| Note: | The list shows only tables and database views that meet the scope protections for field styles. |
| Note: | The value only affects list field styles. To apply field styles on both lists and forms, leave this field blank. |

4. Optional: To add alternative text for a style, **configure the form** to add the alternative text field. For styles like the VIP style icon, you can add alternative text so that screen readers can differentiate between the style and the text.

5. Click **Submit** or **Update**.

**Advanced style selection**

Use a script-based-condition to determine if the system should apply a style to a field in the list.

You can enter a script-based-condition in the **Value** field with the syntax `javascript:<script>`. If the evaluated script returns true, the system applies the style to the list item.

The script has access to the current script object, which allows you to build conditions based on the field values of the current record. For example, this script checks for overdue items based on a field value:

| javascript:gs.dateDiff(gs.now(), current.u_datefield.getDisplayValue(), true) < 0 |

| Note: | In this example, `current.u_datefield` refers to the name of a custom date field. |
The system only supports one `javascript:<script>` entry in a Value field. If you want to enter multiple JavaScript conditions, consolidate the conditions into one statement. For example, instead of this invalid entry:

```javascript
javascript: current.state == "Completed" javascript: current.error_tables > 0
```

Consolidate the conditions into one statement such as:

```javascript
javascript: current.state == "Completed" && current.error_tables > 0
```

Define reference styles

Reference styles are similar to field styles, but apply to any reference to a table. You can define styles that apply to any reference to a table.

Role required: admin

For instance, a reference style on the User [sys_user] table would apply to any reference field on any table that refers to User.

1. Enter `sys_ui_reference_style.list` in the application navigator filter and press Enter.
2. Click **New**.
3. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>If the check box is selected, the style is applied where appropriate.</td>
</tr>
<tr>
<td>Table</td>
<td>The table to which all references will be styled.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The list shows only tables that meet the scope protections for table styles.</td>
</tr>
<tr>
<td>Style</td>
<td>A CSS style to apply.</td>
</tr>
</tbody>
</table>

Managing record numbering

Records in tables can be numbered automatically.

In the base system, several tables are numbered, including Incident, Problem, Change Request, and Knowledge. You can also use these numbers anywhere that script is present, for example to generate watermarks for emails.

Administrators can manage record numbering by navigating to **System Definition > Number Maintenance**. The current number format for a table, including the prefix (such as **INC** for incidents or **CHG** for changes), is stored in a record on the Number [sys_number] table.
You can renumber auto-incremented tables that extend the task table or manage numbering with a database field named **Number this field**. For information about renumbering custom tables, see *KB0538764: Renumbering auto-incrementated custom tables* in the ServiceNow knowledge base.

### Auto-numbering records in a table

**How to set up numbering for a table.**

**Role required:** admin

1. Navigate to **System Definition > Number Maintenance**.
2. Click a table name to view the number record for that table, or click **New**.

   **Note:**
   
   You can only define one number format per table.

3. Define the number format by completing the fields (see table).
4. Click **Submit** or **Update**.

   If an auto-numbered field does not already exist, a new field is automatically created on the table with the following values:
   - Label: Number
   - Name: u_number
   - Default value:

   ```javascript
   javascript:getNextObjNumberPadded();
   ```

   **Note:** This script renumbers records when the **Number of digits** is updated. To use a script that does not renumber records when the **Number of digits** is updated, open the
dictionary entry for the **Number** field and enter the following script in the **Default value** field.

```javascript
g getNextObjNumber();
```

### Table 218: Auto-numbering records in a table

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select a table.</td>
</tr>
<tr>
<td>Prefix</td>
<td>Enter a prefix for every number in the table (for example, <strong>INC</strong> for Incident).</td>
</tr>
</tbody>
</table>
| Number           | Enter the base number for this table (default value is 1000). Record numbers are automatically incremented, and the next number is maintained in the Counter [sys_number_counter] table.  

If you set the base number to a value higher than the current counter, the next record number uses the new base number. Otherwise the next record number uses the current counter. The counter does not reset to a base number lower than itself. To see the current counter, click the **Show Counter** related link.  

| Number of digits | Enter the minimum number of digits to use after the prefix (default value is 7).  

- Leading zeros are added to auto-numbers, if necessary. For example, **INC0001001** contains three leading zeros.  
- The number of digits can exceed the minimum length. For example, if **Number of digits** is **2** and more than 99 records are created on the table, the numbers continue past 100 (such as **INC101**).  

---

**Warning:** Changing this field may update all number values for existing records on a table. Take care when changing this field on a production instance.

---

**Note:**

To change the default values for new number formats, change the **Default value** field on the system dictionary record for the **Number** or **Number of digits** field. These fields are on the **Number** table.
Prepare to left-pad number fields in custom tables

Before you configure left padding of number fields on a custom table or a table that does not extend the task table, you must prepare business rules and script includes.

Role required: admin

1. Duplicate the base system business rule named Pad Numbers: Click **Insert and Stay**. If **Insert and Stay** is not available on your instance, create a new business rule and manually copy all field values from Pad Numbers.
2. Give the business rule a new name and change the script field to read as follows:

```javascript
function padCurrentCategory() {
    var target = new GlideRecord(current.category + "");
    if (!target.isValidField("u_number") || target.isValidField("number"))
        return;
    var nm = new UNumberManager();
    nm.padTableNumbers(current.category, current.maximum_digits);
} Save the changes.
```

3. Duplicate the base system script include named **NumberManager**.
4. Change the following two lines as indicated:

```javascript
105: currentNumber = records.u_number.toString();
117: records.u_number = prefix + currentNumber;
```
5. Name the new script include **UNumberManager** and save your changes.
You can now continue with the process of configuring left-padding and renumbering records.

Configure left padding of a system number in a table

You can configure the left padding of the system numbers on a table. For example, pad the **Number** field on an Incident, Problem, or Change Request.

If you are configuring numbers on a custom table or a table that does not extend the task table, then, before performing the following procedure, you must prepare business rules and script includes. For more information, see **Prepare to left-pad number fields in custom tables** on page 856.

Role required: admin

1. Navigate to the form, then right-click the **Number** field and select **Configure Dictionary**.
2. Enter the following script in the **Default value** field and click **Update**.

```javascript
javascript:getNextObjNumberPadded();
```
3. Navigate to **System Definition > Number Maintenance**.
4. Open the table record.
5. Enter a value in the **Number of digits** field.
6. Click **Update**.

Number padding is applied to both existing and new records.
Prevent numbering gaps

By default, numbers are generated every time a new record is created. When records are created but not saved, a gap in the numbering is created. You can prevent these numbering gaps by generating numbers only when records are saved.

Role required: admin

1. Navigate to System Properties > System.
2. Set the property Assign a task number only upon insert (prevents unused numbers), glide.itil.assign.number.on.insert, to true.
3. Click Save.

Enforcing unique numbering

Although duplicate numbers are rare, numbering does not enforce uniqueness, by default.

To enforce uniqueness, you can:

• Create a before business rule on insert only to check for duplicate values and replace duplicates with the next available number.
• Enable a unique index on the table.

**Note:** While unique indexes ensure data integrity they also prevent any insert involving a duplicate number. This may cause unexpected errors during data entry.
Sample business rule

This sample script can be used as part of a before business rule on insert only to check for duplicate numbers and replace them with the next available number.

```javascript
var curNum = current.number + '';
if(curNum) {
    var recordClass = current.getRecordClassName();
    var gr = new GlideRecord(recordClass);
    gr.addQuery('number', curNum);
    gr.query();
    if(gr.getRowCount() > 0) {
        var newNum = getNextObjNumberPadded();
        gs.addInfoMessage("The number " + current.number + " was already used by another " + recordClass + ". The " + recordClass + " number has been changed to " + newNum);
        current.number = newNum;
    }
}
```

Field spell checking

Spell checking can be added to any journal field, such as a **Comments** box or an HTML editor. Spell checking can be added to any input field, but is not typically done for single line input fields. To enable the spell checking dictionary, install the appropriate language plugin. Localization plugins have the same name format, such as I18: German Translations.

**Note:**
- Field spell checking is not supported in UI16 or UI15.
- The words in the dictionary are part of the language plugin. You cannot add, remove, or modify the dictionary.

Add spell checking to a field

You can add a spell check option to any field in UI11.

Role required: personalize_dictionary

**Note:** Field spell checking is not supported in UI16 or UI15.

1. Navigate to **System Definition > Dictionary**.
2. Filter the list to view **journal** type fields and select a field to spell check.
   An example is the **comments** field in the Task table. This applies spell checking in the specified language to the multi-line **Comments** text box for incidents, problems, and changes.
3. Click the table name to open the Dictionary form.
4. Configure the form and add the **Spell check** check box.
5. Select the check box to enable spell checking in every **Comments** field on the Task table.
6. To use the spell checker in a task record, click the spell check icon ( ) next to the field label.
The spell checker highlights the misspelled words.

Select dictionaries for spell checking

In UI11, specific dictionaries are available for users to check spelling in fields. After you activate a language plugin, you can configure the dictionary to use for languages that don't have a spell check dictionary.

The dictionaries available for spell checking include the following:

- Brazilian Portuguese
- Czech
- Dutch
- English US
- English UK
- Finnish
- French
- German
- Hebrew
- Hungarian
- Italian
- Polish
- Portuguese
- Russian
- Spanish
- Thai

1. Navigate to **System Properties > System Localization**.
2. Select the dictionary to use for the language that does not have a spell checker dictionary.

3. Click **Save**.

Condition count widget

The condition count widget can be activated on condition fields to display a preview of the records that would meet the current set of conditions.
For fields where the condition count widget is activated, the number of records that match the conditions will automatically display. The count refreshes if the field the condition field depends on, such as Table, is changed; if the Table field is left blank, the widget is hidden.

In this example, a condition is run on the Incident table to look for incidents where Category is Network:

![Figure 273: Condition count](image)

The list shows that there are 112 records with Network as its category.

To refresh the preview, click the update count icon ( ).

To view details of the results, click the number of records to open the list view of the results:

![Figure 274: Condition count results](image)

Adding the condition count widget

You can add the condition count widget to a condition field.

Role required: personalize_dictionary

1. Right-click the field label and select Configure Dictionary.
2. Add show_condition_count=true to the Attributes field.
3. Submit.

Reference fields

A reference field stores a reference to a field on another table. For example, the Caller field on the Incident table is a reference to the User [sys_user] table.

When you define a reference field, the system creates a relationship between the two tables. Adding a reference field to a form makes the other fields in the referenced table available to the form.
**Note:** A reference field can refer only to records from one other table. To add a field that can refer to records on any table, use the Document ID element type.

Administrators can create new reference fields and configure several options for reference fields.

**Table 219: Reference field options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display values</td>
<td>Each reference field stores a sys_id for each referenced record in the database, but the sys_id is not shown. The reference field shows the specified display value.</td>
</tr>
<tr>
<td>Decorations</td>
<td>A reference decoration is an icon that appears next to a reference field.</td>
</tr>
<tr>
<td>Reference styles</td>
<td>Reference styles are specialized field styles that control the appearance of reference fields.</td>
</tr>
<tr>
<td>Reference qualifiers</td>
<td>Reference qualifiers restrict the records that are available for reference fields.</td>
</tr>
<tr>
<td>Cascade delete rules</td>
<td>Cascade delete rules specify what should happen to records that reference a record that is deleted.</td>
</tr>
<tr>
<td>Auto-complete</td>
<td>By default, a reference field auto-completes as the user types in the field. Administrators can configure auto-complete settings.</td>
</tr>
<tr>
<td>Reference key</td>
<td>A reference key saves a field other than sys_id as the unique identifier for a reference field.</td>
</tr>
<tr>
<td>Enable dynamic creation</td>
<td>When dynamic creation is enabled, entering a nonexistent value in a reference field creates a new record on the referenced table instead of returning an error.</td>
</tr>
</tbody>
</table>

**Add a reference field**

Add reference fields to a table using the same method as for any other field.

Role required: personalize_form

The related table also appears in the Available Tables list for future form customizations.

1. Open the desired form.
2. Right-click the header and select **Configure > Form Layout**.
3. Use **dot-walking** to locate and select the field in the referenced table that you want to add. It appears as Table name.Field. For example, the caller's email address appears as Caller.Email.
4. Click **Save**.

**Display values**

Reference fields store a sys_id for each referenced record in the database, but the sys_id is not shown.
The reference field shows the display value. For example, an incident's **Assigned to** field stores the sys_id of a particular user, but actually displays the user's name. The following example shows how **Charlie Witherspoon**, which is the display value of a user record, is shown in the **Assigned to** field.

Figure 275: Display value xml

<table>
<thead>
<tr>
<th>Reference field</th>
<th>Value stored in database</th>
<th>Display value field of source table</th>
<th>Value displayed in UI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigned to</td>
<td>46b87022a9fe198101a78787e40d7547</td>
<td>[sys_user.name]</td>
<td>Charlie Whitherspoon</td>
</tr>
</tbody>
</table>

Reference fields show display values in:

- **Lists**
- **Forms**
- **Reports**
- **Auto-complete suggestions**
- **Slushbuckets**

**Select a field as the table display value**

Only one field can be defined as the display value for a table.

Role required: personalize_dictionary

When you set the **Display** value to **true**, a business rule sets the **Display** value to **false** for all other fields on the table. In previous versions, you must manually ensure that no other fields on the table have a value of **true** in the **Display** column.

**Note:** Extended tables inherit the display value of the parent table. Setting a separate display value for the extended table overrides the parent table's display value.

1. Navigate to **System Definition > Dictionary**.
2. Filter on `[Table] [is] [<name of the referenced table>]`.
3. Locate the desired field and set **Display** to **true**.

   For best results, choose a field that is required and unique in each record as the display value field.
**Note:** If you make a field the display field for a table, be sure to translate all values for the field in the Translated Text [sys_translated_text] table into all the languages provided. Display field options left untranslated are not presented by the autocomplete (type ahead) feature.

Reference fields look for the display value in the following order:

1. A field with `display=true` in the system dictionary on the lowest sub-table for extended tables.
2. A field with `display=true` in the system dictionary on the parent table.
3. A field named `name` or `u_name`.
4. The **Created on** field of the referenced record.

**Decorations**

Reference decorations are icons which appear next to the reference field. The reference lookup is always visible and is used to select a record to reference. Other reference decorations appear when a record is selected.

**Reference lookup**

Clicking the reference lookup icon (🔍 in UI16/UI15) displays a list of records on the referenced table in a pop-up window.
Figure 276: Reference lookup list UI16 and UI15
The reference lookup can be rendered in the tree picker format by modifying the dictionary and adding the attribute `tree_picker`.

You cannot customize the label names used in the tree picker. The label names are taken from the values in the table.

*Reference field icon*

When a record is referenced, pointing to the reference icon (自信 in UI16/UI15, 信息 in UI11) displays information about the referenced record.
Figure 278: UI15 reference record

Clicking the reference icon updates the current form and navigates to the referenced record.

Customizing the reference icon

Administrators can use the sys_popup form view to configure the fields in the pop-up that appears when pointing to a reference icon.

Configure the fields in the pop-up for a reference icon

You can use the sys_popup form view to configure the fields in the pop-up that appears when pointing to a reference icon. If the table has no sys_popup view, the pop-up uses the default view.
Role required: personalize_form

1. Navigate to the following URL, substituting the instance name and table name as appropriate:

   `<your instance name>.service-now.com/<table name>.do?sysparm_view=sys_popup`

2. Configure the form to add or remove fields as appropriate.

Configure the pop-up delay for a reference icon

The pop-up appears after a configurable period of time. This default delay is 100 ms, and you can configure a different value

Role required: admin

   Note: Setting the value extremely low can lead to unwanted pop-ups being requested from the server. Additionally, a low setting can make it difficult to drill through the reference icon because the icon is not clickable until the pop-up appears. Conversely, an extremely high setting makes the pop-ups somewhat sluggish.

1. Navigate to System Properties > UI Properties.
2. Modify the property called Record popup delay (milliseconds): (glide.ui.popup.delay).
   The property controls when the request for the pop-up is made; the actual display time depends on the browser.
3. Click Save.

Configure pop-ups on read-only fields

Reference pop-ups and click-throughs are hidden by default if a client script, UI policy, or ACL makes the field read-only. The ability to see or click through to the target record does not depend on whether the reference field is writable. You can change the read-only setting.

Role required: admin

1. Navigate to System Properties > UI Properties.
2. Change the value of the Enable click-through of a reference field when the reference field is read-only. (glide.ui.reference.readonly.clickthrough) property.
   If set to true, the pop-up appears for read-only fields.

   If this system value is set to false, you can override the setting for a specific read-only reference field. Configure the dictionary entry and add the readonly_clickthrough=true attribute.
Configure the related incidents icon
You can configure an icon to appear beside a reference field in a form, such as the **Caller** field, to display related incidents.

Role required: admin

The show related incidents icon (in UI16 and UI15, in UI11) displays other incidents related to the referenced record.

1. In the form, right-click the label for the reference field and select **Configure Dictionary**.
2. Add the `ref_contributions=user_show_incidents` dictionary attribute in the **Attributes** field.
3. Click **Update**.
   - The form reopens and the related incidents icon appears beside the field on the right.

**Configure the show workflow icon**
You can configure an icon to appear beside a workflow field to display the related workflow in the workflow editor.

Role required: admin
The show workflow icon ( ![in UI16 and UI15] in UI16 and ![in UI11] in UI11) opens the workflow in the workflow editor.

1. In the form, right-click the label for the workflow field and select **Configure > Configure Dictionary**.
2. Add the `ref_contributions=show_workflow` dictionary attribute in the **Attributes** field.
3. Click **Update**. The form reopens and the show workflow icon appears beside the field on the right.

**Reference styles**

Reference styles are specialized field styles that control the appearance of reference fields. For more information, see *Define reference styles* on page 853.

**Reference qualifiers**

Use reference qualifiers to restrict the data that is selectable for a reference field.

For example, if you want another record to be able to reference records with the **San Diego** location name, you can create a reference qualifier on the **Location** field of the record. The qualifier specifies that the user can choose only location records with the **City** field set to **San Diego** for the **Location** field.

**Note:** Reference qualifiers are available when using reference lookup (magnifying glass) from forms, applying the qualifier condition and displaying the filtered information in the displayed list. They are not applicable for condition builders.

**Types of reference qualifiers**

The following types of reference qualifiers are available:

- Simple qualifiers provide choice lists for you to specify a reference qualifier condition on the table where the reference field is located. Simple qualifiers only apply when using "type-ahead" functionality on a form; they do not apply to condition builders.
- Dynamic qualifiers allow you to use a dynamic filter to run a query against a reference field without having to enter JavaScript code or query strings.
- Advanced qualifiers provide a text field for you to create either of the following:
  - A static encoded query string, which is a single string that specifies a database query, such as `active=true`.
  - JavaScript code that references script includes or functions in global business rules.

**Note:** To define a reference qualifier on an extended table differently from the reference qualifier on the parent table, use a Dictionary Override.
3. To change views, select **Default view** or **Advanced view** under **Related Links**.

4. In the Reference Specification section, verify that the table already present in the **Reference** field is the correct one, or select another table if necessary.

5. Select the type of qualifier in the **Use reference qualifier** choice list.

6. Configure the qualifier:
   - **Simple**: *Build the condition* using the choice lists.
   - **Dynamic**: *Select a dynamic filter option*.
   - **Advanced**: Enter the encoded query string or the JavaScript that returns a query string in the **Reference qual** field.

7. In the **Reference Specification - Additional Customization** section, configure these options if necessary:
   - **Reference key**: allows you to *identify a field other than sys_ID* to use as the unique identifier for the reference field.
   - **Reference cascade rule**: allows you to *define what happens to a record* if the record it references is deleted.
   - **Reference floats**: *enables the Edit button on related lists* for one-to-many relationships.
   - **Dynamic creation**: lets you determine if the system should *create a new record* when a value for the reference field does not match an existing record. If you select this option, enter a script in the **Dynamic creation script** field that specifies how to create the record.

8. Click **Update**.

**Simple reference qualifier example**

Use simple reference qualifiers when you want to limit the values for a reference field based on other values in the referenced table or related tables.

The base system provides several simple reference qualifiers by default. An example is the reference qualifier on the **Vendor** field on asset forms, such as the Hardware form. The qualifier restricts the companies you can select for this field to only those companies with the **Vendor** field set to **true**.

![Simple reference qualifier example](image)

**Figure 280: Simple reference qualifier**

**Note**: Simple reference qualifiers can have a maximum of thirteen reference qualifier conditions.

**Dynamic reference qualifier example**

The advantage of using a dynamic reference qualifier is that you can create one dynamic filter option and in as many dynamic reference qualifiers as needed.
Use dynamic reference qualifiers when you want to limit the values for a reference field based on *Create a dynamic filter option* on page 80 that uses *Create scripted filters* on page 80.

The base system provides several dynamic filter options by default. An example is the dynamic filter option for the reference qualifier on the **Model ID** field. This field appears on a configuration item form, such as the default **Computer** form. The reference qualifier calls the **CI Model Qualifier** dynamic filter option, which in turn calls the **ModelAndCategoryFilters** script include. This script include refines the reference qualifier based on the class of the CI so that the only options for the model ID are those that belong to the same class as the current CI. For example, only CIs that belong to the Computer class are available in the **Model ID** field on a Computer configuration item form.

Figure 281: Dynamic reference qualifier

**Advanced reference qualifier example**

Use advanced reference qualifiers if you want to enter an encoded query string or make a JavaScript call to a script include that returns a query string.

To create advanced reference qualifiers, enter the query string or JavaScript code in the **Reference qual** field on the Dictionary Entry form.

---

**Note:** As a good practice, make JavaScript calls to functions in a script include instead of a global business rule. See for more information.

**Encoded query strings**

An example of an encoded query string is `vendor=true`, which returns all companies that are designated as vendors. Entering this string is the same as using the condition builder as shown in the example for the simple reference qualifier.
JavaScript calls

JavaScript calls.

An example of a JavaScript call is the following code:

```
javascript:new myScriptInclude().my_refqual()
```

This code calls a function named `my_refqual()` in a script include named `myScriptInclude()`. The function must return a query string that can filter the options available on a reference field.

**JavaScript example: Limiting the Assigned to Field by Users with a Specified Role**

This example shows how to restrict an incident's Assigned to choices to only the users with the itil_admin role. You could also change `itil_admin` to any other role on a reference field that refers to the User table.

1. Open an incident.
2. Right-click the Assigned to field and select Configure Dictionary.
3. In the Reference qual field, enter
   ```javascript:
   "sys_idIN"+getRoledUsers("itil_admin").join(',').
   ```
4. Save the record.
5. To see the base-system business rule that this JavaScript code calls, navigate to System Definitions > Business rules.
6. Open getRoledUsers.
7. The business rule uses the following JavaScript code:

   ```javascript
   // Return an array of sys_ids of the users that have at least one role
   // optional parameters allow the exclusion (NOT IN) of some roles or//
   // look for specific roles (IN)
   //
   // optional: queryCondition - 'IN' or 'NOT IN'
   // optional: roleList - a comma separated list of role names
   //
   function getRoledUsers(queryCondition, roleList){
     var roleListIds;
     if(queryCondition && roleList){
       roleListIds = getRoleListIds(roleList);
     }

     var users ={};
     var gr =new GlideRecord('sys_user_has_role');
     if(roleListIds){
       gr.addQuery('role', queryCondition, roleListIds);
     }
   }
   ```

© 2017 ServiceNow. All rights reserved.
JavaScript example: Limiting the Assigned to field

This example shows how to restrict an incident's **Assigned to** choices to only the users with the itil_admin role.

You could also change itil_admin to any other role on a reference field that refers to the User table.

1. Open an incident.
2. Right-click the **Assigned to** field and select **Configure Dictionary**.
3. In the **Reference qual** field, enter
   ```javascript
   sys_id IN getRoledUsers("itil_admin").join(",")
   ```
4. Save the record.
5. To see the base-system business rule that this JavaScript code calls, navigate to **System Definition > Business Rules**.
6. Open **getRoledUsers**.

The business rule uses the following JavaScript code.

```javascript
// Return an array of sys_ids of the users that have at least one role
// optional parameters allow the exclusion (NOT IN) of some roles or
// look for specific roles (IN)
//
// optional: queryCondition - 'IN' or 'NOT IN'
// optional: roleList - a comma separated list of role names
//
// function getRoledUsers(queryCondition, roleList) {
// var roleListIds;
// if (queryCondition && roleList) {
//  roleListIds = getRoleListIds(roleList);
// }
// var users = {};
// var gr = new GlideRecord('sys_user_has_role');
// if (roleListIds) {
//  gr.addQuery('role', queryCondition, roleListIds);
// }
// gr.query();
// while (gr.next()) {
//  users[gr.user.toString()] = true;
// }
// var ids = [];
// for (var id in users)
//  ids.push(id);
// return ids;
// }

// get sys_id's for the named roles
function getRoleListIds(roleList) {
```
var(ids)[];var gr = new GlideRecord('sys_user_role');
gr.addQuery('name', 'IN', roleList);
gr.query();
while (gr.next()) {
    ids.push(gr.sys_id.toString());
} return ids;
}

JavaScript example: Constraining the assignment group field

This example shows how to restrict an incident's Assignment group choices to only the groups that contain the user already specified in the Assigned to field.

1. Open an incident.
2. Right-click the Assignment group field and select Configure Dictionary.
3. If the form appears in Default view the Related Links list, click Advanced view.
4. In the Use reference qualifier field, ensure that the Advanced option is selected.
5. In the Reference qual field, enter javascript:new BackfillAssignmentGroup().BackfillAssignmentGroup().
6. Save the record.
7. Navigate to System Definitions > Script Includes.
8. Click New.
9. Create a script include with the following JavaScript code:

var BackfillAssignmentGroup = Class.create();
BackfillAssignmentGroup.prototype = {
    initialize: function() {
    },

    BackfillAssignmentGroup:function() {
        var gp = '';
        var a = current.assigned_to;

        //return everything if the assigned_to value is empty
        if(!a) return;

        //sys_user_grmember has the user to group relationship
        var grp = new GlideRecord('sys_user_grmember');
gr.addQuery('user', a);
gr.query();
while(grp.next()) {
    if (grp.length > 0) {
        //build a comma separated string of groups if there is more than one
        gp += (',' + grp.group);
    } else {
        gp = grp.group;
    }
}

    // return Groups where assigned to is in those groups we use IN for lists
    return 'sys_idIN' + gp;
},
type: 'BackfillAssignmentGroup'
The next time you create an incident, select a user in the **Assigned to** field. Then click the **Assignment group** lookup icon. Only the groups that contain the user you just selected appear.

For example, if Bob Smith belongs to the Database group and the Networking group and you assign an incident to Bob, the only options you can select for the assignment group are Database and Networking.

**Related lists and reference qualifiers**

Whenever you edit a reference field from a related list, it might be necessary to know which related list the reference field is on in order to properly build the reference qualifier for the field.

This occurs when the same field appears on different related lists and the context is necessary to know how to properly qualify the reference values.

To do this, configure the list control for the related list and fill in the **List edit tag** with any tag you choose. This tag value will be available to the advanced reference qualifier function as a variable named `listEditRefQualTag`. For example, your advanced reference qualifier script include can look like this:

```javascript
// Advanced reference qualifier on the CI Relationship Child field that takes into account the related list that we are editing the child field on, if the field is being edited from a tagged related list.

cmdb_rel_ci_child_refQual:function(){
  if(listEditRefQualTag =="application")return"sys_class_name = cmdb_ci_appl";
  if(listEditRefQualTag =="database")return"sys_class_name = cmdb_ci_database"
}
```

**Use the INSTANCEOF operator**

Use the INSTANCEOF operator in a reference qualifier to shorten or simplify a complex class qualifier.

For example, you can use the INSTANCEOF operator for a reference field to the `cmdb_ci` table to specify that you want all subclasses of a class included in the results. The following reference qualifier returns all servers, including Linux, UNIX, Windows, and so on, because each of those subclasses extend the `cmdb_ci_server` class:

```
sys_class_nameINSTANCEOFcmdb_ci_server
```

Likewise, the following reference qualifier can be similarly simplified:

```
u_active=true^sys_class_name=cmdb_ci_acc
^ORsys_class_name=cmdb_ci_computer
^ORsys_class_name=cmdb_ci_server
^ORsys_class_name=cmdb_ci_win_server
^ORsys_class_name=cmdb_ci_unix_server
^ORsys_class_name=cmdb_ci_linux_server
^ORsys_class_name=cmdb_ci_appl
^ORsys_class_name=cmdb_ci_netgear
```

If you use the INSTANCEOF operator, the reference qualifier is as follows:

```
u_active=true^sys_class_name=cmdb_ci_acc
^ORsys_class_name=cmdb_ci_computer
^ORsys_class_nameINSTANCEOFcmdb_ci_server
^ORsys_class_name=cmdb_ci_appl
^ORsys_class_name=cmdb_ci_netgear
```
Troubleshooting

If you receive a **Bad Gateway Error** in Internet Explorer when searching in a reference lookup window, then you cannot use `ref_qual_elements=*` in the advanced reference qualifier.

**Reference qualifiers for service catalog variables**

Service Catalog reference variables restrict the available values for choice-based variable types.

Reference Qualifiers

Service Catalog reference variables can use advanced reference qualifiers to restrict the available values for choice-based variable types based on the result of a script or other variables on the catalog item.

For example, you could create a **Location** reference variable with a reference qualifier that restricts the available location choices to those associated with groups in which the **Requested For** user is a member.

**Note:** This functionality requires knowledge of scripting.

Reference Qualifier Uses

The `ref_qual_elements` attribute can be used to specify dependent fields in the following cases:

- Catalog Item ordering screen
- Requested Item form
- Catalog Task form

In the JavaScript function called by the advanced reference qualifier, the current record (`current`) is in scope, allowing access to additional information. For example:

- Variable information for the ordered item is available with `current.variables.variablename`.

Filter names example

Following is an example script of a function.

The function defines an advanced reference qualifier for a user (`sys_user`) reference variable that filters the available user records to those where **Name** contains the value of a variable named `textmatch` on the same item:

```javascript
function getBlackberryUsers () {
  var answer = '';
  var includes = current.variables.textmatch;
  var usr = new GlideRecord('sys_user');
  usr.addQuery('name', 'CONTAINS', includes);
  usr.query();
  while (usr.next()) {
    if (answer.length > 0) {
      answer += (',' + usr.sys_id);
    } else {
      answer = '' + usr.sys_id;
    }
  }
  return 'sys_idIN' + answer;
}
```

Configure cascade delete rules

When a record is deleted, there are different options for how the deletion will affect records that reference the deleted record. You can configure what happens to records that reference a record when that record is deleted.

Role required: personalize_dictionary
For example, if you delete a user record that is referenced in the **Caller ID** field on several incident records, you can configure what happens to those incident records. By default, the references are cleared, so the incident records are maintained with an empty **Caller ID** field.

1. Navigate to a reference field on a form.
2. Right-click the field label and select **Configure Dictionary**.
3. Under **Related Links**, click **Advanced view**.
5. In the **Reference cascade rule** field, select one of the following options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear or -- None --</td>
<td>Deleting a record clears references (default option).</td>
</tr>
<tr>
<td>Delete or Cascade</td>
<td>Deleting a record also deletes all referencing records. For example, when a user record is deleted, any incidents assigned to the user are also deleted.</td>
</tr>
<tr>
<td><strong>Caution:</strong> Use this method with caution.</td>
<td></td>
</tr>
<tr>
<td>Restrict</td>
<td>Deleting a record is restricted unless there are no references to the record. For example, prevent the user record from being deleted if any incident includes a reference to the user. This option has no effect for tables with m2m relationships, meaning that in this example, if the tables have an m2m relationship, the user record is deleted.</td>
</tr>
<tr>
<td>None</td>
<td>Deleting a record does not change records that reference the record.</td>
</tr>
</tbody>
</table>

6. Click **Update**.

**Auto-complete for reference fields**

By default, a reference field auto-completes as the user types in the field. Administrators can configure additional auto-complete options. A user must have table-level read permission on the referenced table for auto-complete to display any options.

```
<table>
<thead>
<tr>
<th>Caller:</th>
</tr>
</thead>
<tbody>
<tr>
<td>joel</td>
</tr>
<tr>
<td>Location:</td>
</tr>
<tr>
<td>Joe Employee</td>
</tr>
</tbody>
</table>
```

**Figure 283: Auto complete**

**Dictionary attributes for auto-completion of reference fields**

There are dictionary attributes that are specific to reference fields and that determine auto-complete behavior.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref_auto_completer</td>
<td>Specifies the name of the client-side JavaScript class that creates the drop-down auto completion choices. Valid class values include:</td>
</tr>
<tr>
<td></td>
<td>• AJAXReferenceCompleter: Displays matching auto-complete choices as a drop-down choice list. The list only displays the reference table’s display value column. Reference fields automatically use this class if there is no other auto-completion class specified.</td>
</tr>
<tr>
<td></td>
<td>• AJAXTableCompleter: Displays matching auto-complete choices as rows in a table. The table displays the reference table’s display value column and any columns listed in the ref_ac_columns attribute.</td>
</tr>
<tr>
<td></td>
<td>• AJAXReferenceChoice: Displays matching auto-complete choices as a drop-down choice-list. The list only displays the reference table’s display value column. Furthermore, the list only displays up to 25 matching choices. If there are more than 25 auto-complete choices, the reference field instead displays the choices with the AJAXTableCompleter class.</td>
</tr>
<tr>
<td>ref_ac_columns</td>
<td>Specifies the list of reference table columns to display. Separate column names with a semi-colon. For example, ref_ac_columns=user_name;email;sys_created_on allows auto-complete to match text from the user_name, email, and sys_created_on columns.</td>
</tr>
<tr>
<td>ref_ac_columns_search</td>
<td>Enables auto-complete to match text in the columns listed in the ref_ac_columns attribute. Set this attribute to true to enable auto-complete to match text in all reference field columns. By default (or when this attribute is false) auto-complete only matches text in the display value column.</td>
</tr>
<tr>
<td>ref_ac_order_by</td>
<td>Specifies the reference table column that sorts the auto-completion choices. For example, ref_ac_order_by=name sorts the auto-completion choices alphabetically by name.</td>
</tr>
</tbody>
</table>

Administrators can also set a user preference to use a contains auto-complete search.

**Auto-complete UI features**

UI Features

The AJAX table completer class has a number of UI improvements:
• The table completer always displays the number of records the auto-complete query finds.
• The table completer highlights the entire selected row by changing the color of the background and text.
• The table completer lists a value for every column.
  • The first time a value appears in a column, the table completer displays it in black text.
  • The table completer displays subsequent duplicate values in grey text. Previously, the table completer displayed an empty cell in a column containing a duplicate value.

Figure 284: Selected row highlight

Set the ref_auto_completer=AJAXTableCompleter dictionary attribute to use these improvements.

Define auto-complete attributes for all references to a table
A field inherits and uses the reference table's auto-complete attributes unless the field has its own value for the same attributes. You can define the attributes for references to a table, and it affects every form that references that table.

Role required: personalize_dictionary
A field-level attribute overrides a table-level attribute of the same name. If a field uses different reference attributes from those that are defined for the reference table, then the field uses both sets of attributes.

Use these steps to define auto-complete attributes for all fields in a table that do not already have their own auto-complete attributes. This example describes how to define auto-complete attributes for all references to the User [sys_user] table.

Note: A field's auto-complete attribute value supersedes a table's auto-complete attribute value. This means that any existing field-level value for an auto-complete attribute supersedes any value the administrator applies to the auto-complete attribute from the reference table.

1. Navigate to a form of the target table, such as User Administration > Users.
2. Right-click the column header and click Configure > Dictionary.
3. Select the row that does not list a column name.
   This row is typically the first row in the list. For example, select the first sys_user link.
5. In the Attributes field, enter a comma-separated list of auto-complete attributes you want to apply to all fields in the table.
   For example, to display the user's department with all references to the sys_user table, enter:
   ```
   ref_auto_completer=AJAXTableCompleter,ref_ac_columns=department,ref_ac_order_by=department
   ```
6. Click Update.
To test the new auto-complete attributes, open a form that references the User [sys_user] table, such as an open incident. Enter a single character in the **Assigned to** field. The auto-complete options now include both the user name and department.

Remove the display value column
You can remove the display value column from a reference field by setting the ref_ac_display_value attribute to false.

Role required: personalize_dictionary
This causes the reference field to remove the display value column and only display the columns listed in the ref_ac_columns attribute. This feature requires the use of the AJAXTableCompleter class and the ref_ac_columns, ref_ac_columns_search, and ref_ac_display_value attributes.

**Note:** Auto-complete cannot match text from additional columns when the reference field is a product of the ui_reference UI macro. This means any auto-complete action against a selector, such as the Impersonate User list, can only match text against the display value.

This example describes how to remove the display value column from references to the User [sys_user] table and replace it with references to the first_name and last_name columns.

1. Navigate to a form of the target table, such as **User Administration > Users**.
2. Right-click the form header and click **Configure > Dictionary**.
3. Select the row that does not list a column name.
   This row is typically the first row in the list. For example, select the first **sys_user** link.
4. Under **Related Links**, click **Advanced view**.
5. In the **Attributes** field, add the ref_auto_completer, ref_ac_columns, ref_ac_columns_search, and ref_ac_display_value attributes.
   For example, to hide the display value column and only display the user’s first and last names enter the following.
   ```
   ref_auto_completer=AJAXTableCompleter,ref_ac_columns=first_name;last_name,ref_ac_columns_search=true,ref_ac_display_value=false
   ```
6. Click **Update**.

To test the new auto-complete attributes, open a form that references the User [sys_user] table, such as an open incident. Enter a single character in the **Assigned to** field. The auto-complete options now hide the display value column (user_name) and only display the first_name and last_name columns.
Improve auto-complete queries
By default, all reference fields use a *starts with* query to search for matching text in the reference table. This prevents auto-complete from executing inefficient *contains* queries every time a user searches a reference field. You can require all reference fields to use a *starts with* query.

Role required: admin

The following example illustrates a *contains* query. Note that the letter "d" appears anywhere in the user's first or last name.


This procedure describes how to change the glide.ui.ref_ac.startswith system property to always use a *starts with* query.

1. In the navigation filter, enter `sys_properties.list` and press the Enter key.
2. Select the `glide.ui.ref_ac.startswith` property.
   
   To search for the property, enter `*startswith` in the Go to search filter for the Name column.
3. In the Value field, replace `false` with `true`.
   
   **Note:** Setting the `glide.ui.ref_ac.startswith` system property to `true` overrides any existing autocomplete.contains settings in both user and system level preferences. This property changes the autocomplete query method for all users regardless of preferences.
4. Click Update.
5. Test the change by opening a record with a reference field and entering a character in it, as illustrated in the example below.
Configure auto-complete to match text from any reference field

By default, auto-complete only matches text in the display value column. You can configure a reference field to match text from any additional column the reference field displays.

Role required: personalize_dictionary

You can add the ref_ac_columns_search attribute to enable auto-complete to match text in any column listed in the ref_ac_columns attribute. Set the ref_ac_columns_search attribute to true to match text from all reference field columns. By default (or when this attribute is false) auto-complete only matches text in the display value column.

1. Right-click the label of a reference field.
2. Select Configure Dictionary from the choice list.
4. In the Attributes field, add the desired auto-completion attributes.

For example, these attributes add the department field to the caller list and sort callers by their department:

ref_auto_completer=AJAXTableCompleter,ref_ac_columns=department,ref_ac_order_by=department
Figure 285: Auto-complete attribute

5. Click **Update**.

The following example describes how to set the **Configuration Item** field display the CI class names from auto-complete choices for the Configuration item [cmdb_ci] table.

```plaintext
ref_auto_completer=AJAXTableCompleter,ref_ac_columns=sys_class_name,ref_ac_order_by=sys_class_name,ref_contributions=task_show_ci_map;ci_show_incidents
```

**Note:** The ref_contributions attribute controls the icons that appear next to the reference field.
Display a reference field as a choice list
You can display a reference field as a choice list instead of opening a lookup window.

Role required: personalize_dictionary

1. In the form, right-click the label for the reference field and select Configure Dictionary.
2. In the Choice List Specification section, select one of the following options in the Choice field.
   - Dropdown with --None--
   - Dropdown without --None--
3. Right-click the form header and click Save.
5. In the Attributes field, add the following attribute, separated from other attributes by a comma.

   "ref_auto_completer=AJAXReferenceChoice"

6. Click Update.
   The form reopens, with the reference field as a choice list.
Note:
When a reference field is displayed as a choice list, the maximum number of choices is limited by the glide.ui.max_ref_dropdown property (default value is 25). Users with the personalize_dictionary role can modify this property, but should test any modifications for performance and usability impact.

Users with the personalize_dictionary role can also override the property value for a specific field by adding the max_ref_dropdown attribute to the field's dictionary entry.

If a reference qualifier returns more choices than the limit for the field, then the field is displayed as a reference field, not as a choice list.

This works on reference fields with advanced reference qualifiers.

Enable contains auto-complete searches
By default, the reference auto-complete uses a starts with search. A user preference can be created to implement a contains search.

Role required: admin
1. Disable the glide.ui.ref_ac.startswith system property.
   For more information, see Improve auto-complete queries on page 881.
Note: Setting the glide.ui.ref_ac.startswith system property to true overrides any existing "autocomplete.contains" settings in both user and system level preferences. This property changes the auto-complete query method for all users regardless of preferences.

2. Navigate to User Administration > User Preferences.
3. Select the preference "'<referenced table>.autocomplete.contains'".
4. Set the value field to true.

![User Preference Configuration](image)

5. Click Update.

Log out and log back in to display the updated search.

Wildcards in reference auto-completes

Wildcard searches use the auto-complete functionality.

Use an asterisk in the reference field for wildcard searches.

![Assigned to:](image)

Figure 286: One asterisk wildcard search

If two asterisks are entered, a list of available records display in the auto-complete suggestions.
Define the reference key

By default, reference fields store the sys_id of the record in the database.

Role required: personalize_dictionary

By defining a reference key, you can identify a field other than sys_id to use as the unique identifier for the reference field. The value of the reference key field, instead of the sys_id, is stored in the database for that reference field.

1. Navigate to System Definition > Dictionary.
2. Open the field record (for example, resolved_by on the Incident table).
3. In the Reference key field, enter a field name on the referenced table (for example, email on the sys_user table).

   Note: Always choose a field from the referenced table that is both required and unique.

4. Click Update.
Enable dynamic creation for reference fields

When dynamic creation is enabled, entering a nonexistent value in a reference field creates a new record on the referenced table instead of returning an error.

Role required: personalize_dictionary

By default, a user must enter a value in a reference field that matches an existing record in the table that the reference field refers to. For example, the Caller field in an Incident must have a value that is an existing user. You can enable dynamic creation to create a new record on the referenced table when a user enter a nonexistent value in a reference field instead of returning an error.

1. Right-click the field label in the form and select Configure Dictionary.
2. Populate the following fields (you may need to configure the Dictionary form):
   - **dynamic_creation**: Select the check box.
   - **dynamic_creation_script**: Enter a script that dynamically creates the record.
3. Click Update.

Examples:

You could use the following **dynamic_creation_script** to create a record on the referenced table.

```javascript
current.name = value;
current.insert();
```

You could create a script include named **MyUserReferenceCreator** with the following contents:

```javascript
var MyUserReferenceCreator = Class.create();
MyUserReferenceCreator.prototype = {
    initialize: function() {
    },
    create: function(current, value) {
        current.name=value;
        return current.insert();
    },
    type: 'MyUserReferenceCreator'
}
```
When the script include is created, the following `dynamic_creation_script` generates a new location for an invalid reference field value:

```javascript
new MyUserReferenceCreator().create(current, value);
```

Recent selections

Reference fields store a list of each user's recent selections to allow users to quickly select past values when filling in a reference field.

By default, the system stores up to 15 selections from a reference field for each user in the Recent Selection [sys_ui_recent_selection] table. Users can see the recent selections list by selecting an empty reference field.

**Note:** The system does not store recent selections for service catalog reference variables.

![Recent selections](image1)

**Figure 288: Recent selections**

The system uses auto-complete to filter the list of recent selections to match values the user enters.

![Recent selections filtered](image2)

**Figure 289: Recent selections filtered**

The system adds a Recent Selection record whenever a user inserts or updates a reference field value. Administrators can control the number of recent selections the system displays with the `glide.xmlhttp.max_choices` system property. Setting the property to zero disables recent selections.

**Note:** This property also controls how many entries the system displays in choice lists.

Create a document ID field

You can create document ID fields to reference any record on any table.

Role required: personalize_dictionary
In comparison, a reference field references a record on a specific table. To reference records from any table, two fields need to work together—one to store the table reference and another to store the record reference.

You can select the document ID type when creating a new field.

1. Navigate to the form view for the table.
2. Right-click the header and select Configure > Form Layout.
3. Create a field to store the table name and click Add.
   - Name: any label (for example, Model table)
   - Type: String
4. Create a field to store the record reference and click Add.
   - Name: label for your document ID field (for example, Model ID)
   - Type: Document ID
5. Click Save.
6. Right-click the form header and select Configure > Dictionary.
7. Open the dictionary entry for the document ID field.
8. Under Related Links, click Advanced view.
9. In the Dependent field, enter the column name of the table reference field (for example, u_model_table).
10. Optional: In Attributes, add the show_all_tables dictionary attribute to display system tables.
11. Click Update. When users click the reference lookup for the document ID field, a dialog appears that allows them to select the table and then the record. The sys_id of the selected record is stored in the document ID field and the table name is stored in the table reference field.
12. Optional: Configure the form to remove the table reference field.

![Select the document dialog](image)

Field normalization

Field Normalization includes two features: normalization and transformation.

- **Normalization** forces the platform to convert different forms of the same field value to a single, accepted value automatically. By forcing a field to use a simple, recognizable description for multiple variations of the same thing, normalization can eliminate duplicate records and make searches easier. In addition to reconciling different forms of the same value in fields, normalization can be configured to adjust queries automatically to return normalized results.
- **Transformation** enables an administrator to transform raw field input into standardized values that are more meaningful to an organization. An example of a standardized value might be to round RAM size in configuration items to a whole number, such as 4000 MB instead of 4112 MB. Transformations are controlled by parameters and conditions and can be configured to return transformed values in queries.
Activate the field normalization plugin

The Field Normalization plugin is published and can be activated by a user with the admin role.

*Activate a plugin* on page 1233

Installed with field normalization

What components are installed with Field Normalization.

The following preferences ([Field Normalization > Administration > Preferences](#)) control features for normalization and transformation:

<table>
<thead>
<tr>
<th>Preference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable field normalization</td>
<td>This preference enables or disables the Field Normalization functionality, including transformations.</td>
</tr>
<tr>
<td>Enable field normalization auditing</td>
<td>This preference enables or disables auditing of field normalization tables.</td>
</tr>
<tr>
<td>Logging</td>
<td>Select the maximum level of logging detail desired. For example, select Error to log only errors and Information to log errors, warnings, and information.</td>
</tr>
<tr>
<td>Decoration URL</td>
<td>Enter the URL for the help/information link associated with the Field Normalization decoration. This link opens the page presented to users who do not have rights to see the normalization or transformation record. The default link opens a ServiceNow Wiki page with a general explanation of Field Normalization. The purpose of this page is to explain to normal users why a field value they entered was changed automatically.</td>
</tr>
<tr>
<td>Restrict decorations to roles</td>
<td>Use this preference to define the user roles that can view the reference icon (decoration) adjacent to a normalized field. This decoration displays transformation and normalization records to users with the normalizer role. For all other users, it opens the URL specified in the Decoration URL preference. If this preference does not specify any roles, then all roles can see the decoration.</td>
</tr>
</tbody>
</table>
The following business rules were created for this plugin:

- Ensure Rules Application Job
- Ensure Transform Application Job
- Ensure Rules and Alias Jobs
- Ensure Transform Application Job
- Ensure Decoration Attribute
- Cleanup on field or table change
- Prevent duplicates in class hierarchy
• Handle potential duplicates
• Run job
• Ensure Pending Value Collection Job
• Ensure name changer job
• Flush forms when activating

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>normalizer</td>
<td>Person authorized to manage field normalization and transformation.</td>
</tr>
<tr>
<td>normalization_tester</td>
<td>Person authorized to create test records for normalizations and transformations. Only records opened by users with this role are available for testing normalization and transformation rules in Test mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldNormalizationAjax</td>
<td>Provides AJAX services for the Field Normalization plugin.</td>
</tr>
<tr>
<td>FNEnsureJob</td>
<td>Ensures that a job record exists for a particular extant data job target.</td>
</tr>
<tr>
<td>FNExtantDataJobChoices</td>
<td>Generates a list of tables for use in document_id field of the fn_extant_data_normalization table.</td>
</tr>
<tr>
<td>FNExtantDataJobUtil</td>
<td>Determines whether an extant data job is allowed to run.</td>
</tr>
<tr>
<td>FNFields</td>
<td>Returns valid fields for normalization.</td>
</tr>
<tr>
<td>FNTransformChooserUtil</td>
<td>Provides support for the fn_transform_chooser UI page.</td>
</tr>
<tr>
<td>Position</td>
<td>Finds a position within a given string.</td>
</tr>
<tr>
<td>Round</td>
<td>Rounds numbers with various rounding modes and intervals.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aliases</td>
<td>Creates the slushbucket with pending values for choosing aliases for a normal value.</td>
</tr>
<tr>
<td>fn_transform_chooser</td>
<td>Displays available transforms. Invoked by the New button on the fn_transform related list.</td>
</tr>
</tbody>
</table>
Field type behavior

Field types determine which fields are available for normalization and which transform categories are used.

- Normalization: Field types determine which fields are available for normalization. When a user creates a new normalization record, the only fields that appear in the choice list are those that match the defined normalization field types for the table selected. For example, in the base system, users see only string fields and not date or integer fields.

- Transformation: Field types determine which transform categories are used. Each category contains a list of transformation definitions that is presented to the user when a new transform is created. The default associations are appropriate for most transforms, but can be reconfigured to suit an organization's business needs.

Normalization field types must match the system field types (from the dictionary). The default field types available depend upon the platform version.

Create a field type

Most field types in the dictionary are available for normalization. Some fields that cannot be normalized are Table and Field (which have hard-coded values) and any reference fields.

To normalize or transform a value in a reference field, apply the processing to the field in the target table.

1. Navigate to Field Normalization > Administration > Field Normalization.
2. Click New in the record list.
3. Enter a Name for the field type that clearly describes the type in the dictionary. This value is for reference only and is not used in any processing. For example, you might enter IP Address for the field type of ip_address.
4. Enter the Type from the dictionary.
5. Select the appropriate check box to use this field type to normalize or transform fields.
6. Right-click in the header bar and select Save from the choice list. The Transform Categories Related List appears.
7. If this field type is being used for transforms, click Edit to associate an existing Transform Category with this field type.

Note: If you create a custom field type that is used for normalizations only, a link to a transform category is not necessary.

The relationship of a field type to a category, and the category to a list of transformation definitions, is completely configurable.
Normalize a field

Field normalization converts different varieties of the same field value, to a single, preferred value.

When a process or a user enters a value in a normalized field, the platform determines if that value can be replaced with a normal value. By forcing a field to use a simple, recognizable description for multiple variations of the same value, normalization can eliminate duplicate records and make searches easier. Normalization also automatically adjusts queries to return normalized results and normalizes values in scripts.

Field normalization is available for every field in the platform, with the exception of reference fields. However, some fields are more likely to be normalized than others, such as the **CPU type** field. Another obvious candidate for normalization is the **Name** field in the Company table. Company names might appear in records in several different forms (for example, Inc., Incorporated, Corporation). Normalized fields can eliminate confusion by providing logical replacements for ambiguous values.
What field normalization does
Records for things like devices and companies are brought into the ServiceNow platform by manual entry, imports, and Discovery.

Depending on how it is introduced into the database, a field value might appear in several different forms. For example, the CPU Type field on a computer CI form might display any of the following names for the same type of CPU, depending on the source of the entry:

- Xeon L3350 or L3350 (manual entry)
- Intel Xeon 5.4.554 (created by Discovery)
- E3350 (Intel) 4.5.2234 (imported value)

The lack of a normalized CPU type field in this example could result in duplicate CMDB records for each variation in the field value. Different CPU types also present a problem for reporting. To report accurately on all computers that use a Xeon processor, an administrator must know all the possible permutations of the name and construct a very complex query. To prevent these issues, an administrator can normalize the original values using one of two methods:

- Aliases: Aliases map all the variations of the name manually to one normal value. Use aliases for short lists of name variants. During processing, the platform looks here first when determining how to normalize a field. If no aliases are defined, the platform searches for rules that apply.
- Rules: Write a rule to associate large numbers of variant names to a normalized value automatically by using standard operators, such as begins with, starts with, or contains. Rules and aliases can be combined to normalize a field. Make sure to test your rules before applying them to all the existing records in the database.

In our example, aliases are sufficient for converting the possible variations of the Intel Xeon CPU type into one, normalized value, such as Xeon. When a recognized variant is entered in a field, the platform automatically replaces the variant with the normalized value. If properly configured, field normalization also affects the search results from a filter in a record list. In our example, an entry of L3350 in a filter returns a list of Xeon CPUs, if that variation of CPU type was normalized.

Scripting and normalization

Scripts that update records or insert records into the database (GlideRecord) are normalized automatically when field normalization is applied. For example, if a script to insert a CI record contains a CPU type of Xeon L3350, the script is normalized to insert the CI with a CPU type of Xeon instead. Scripts that query the database for normalized field values (using the conditions of equals or not equals) can be configured to return the normal value (such as Xeon) rather than the original (raw) value.

Normalized queries

An administrator can configure normalization to apply to queries issued against normalized fields in lists. Select the Normalize query check box on the Normalization form to enable this functionality. In a list containing normalized values, create a filter using the original (raw) value for the normalized field in the query condition.
Figure 291: Normalized query example

The filtered list returns records with the normal value substituted for the raw value. However, the breadcrumbs for the filter display the original query conditions.

Figure 292: Normalized query results

*Data jobs created by field normalization*

Field Normalization creates four data jobs during the normalization process to collect or change data.
All jobs except **Pending value collection** are executed manually. Data jobs have a roll back feature that allows you to undo normalization at different stages before committing changes that affect the entire database.

- **Pending value collection**: This data job is created when a field normalization record is submitted. This job runs automatically to collect all the values from the database for the field to be normalized. This data job does not modify the database.
- **Normal value change**: This data job is created when an existing normal value that is used to normalize a field is changed. When the job is run, the platform updates all the normalized fields in the database that use the new normal value.
- **Alias application**: This data job is created when an alias is created. When the job is run, all field values in the database containing the alias are normalized.
- **Rule application**: This data job is created when a rule is created. When the job is run, all field values in the database that match this rule are normalized.
- **Coalesce to normal**: This data job coalesces data from existing records containing a normalized field value into a single record that uses a normal value.

### Create the normalization record

Regardless of the normalization method selected, all field normalization requires a list of existing variants and a normal value that is configured to replace these variants in forms and in queries.

If you want to also show the original (raw) input in a field on a form after it has been transformed, create a raw field.

Normalize a field by selecting aliases for a normal field value or by creating rules that use condition statements to match field variants with a normal value.

**Note:** Users must have the normalizer role to create and manage normalization records.

### Special Cases

- Reference fields cannot be normalized. To normalize values appearing in reference fields, normalize the field in the target table. Examples of this would be normalized values for the Name fields in the Company [core_company] and Location [cmn_location] tables, which are referenced by other tables in the platform. The normalized names are available to all fields that reference these tables.
- Fields in a choice list can be normalized if they are of a type string.

1. **Activate** the Field Normalization plugin.
2. Navigate to **Field Normalization > Configurations > Normalizations**.
3. Click **New**.
4. Create a normalization record.

### Table 226: Normalization record fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unique name for this normalization record. This value is for reference only and is not used in any processing.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the ServiceNow table for the field being normalized.</td>
</tr>
<tr>
<td>Field</td>
<td>Select the field to normalize.</td>
</tr>
</tbody>
</table>
The three available modes are Off, Test, and Active. All normalization records are created in the test mode by default. If you are planning to select aliases for your normal values, change the Mode to Active. If you intend to normalize the field using rules, be sure to leave this record in the Test mode. To disable this normalization, switch the mode to Off.

Select this check box to apply the field value normalized by this record to all queries involving this field. Queries formed with the raw (original) field value return records displaying the normal value. Queries issued by a script using the conditions equals or not equals return normalized values. See Normalized Queries for details.

Select this check box to reset all references in the database to records containing an alias field value to a single record using a normal value. See Coalesce Normal Values for details.

Select the field to use to display the original (non-normalized) input values on a form in which a field value has been normalized. For the selection to appear in the drop-down list, add a custom field to the form for the table selected. For instructions on adding a field for raw data, see Creating a Raw Field.

After the normalization record is submitted, the platform runs the Pending value collection job in the Data Jobs related list automatically to gather all the current values (Pending Values) for the field being normalized.
Click the **Pending Values** related list to view values for the normalized field that have been entered manually, imported into the platform, or created by Discovery.
**Roll back data jobs**

How to roll back data jobs.

To roll back data gathering, select the job in the **Data Job** related list, and then click the **Rollback** related link in the Data normalization jobs form.
Figure 293: Normalize extant data 2

The values in the **Pending Values** Related List are removed and the data job **State** indicates that it was rolled back.
Create a normal value
A normal value is a simplified, generic value for a field that replaces all the possible variants of that value that exist in the database.

Normal values should be clear and unambiguous.

After the platform runs the data job, the Pending Values related list on the Data normalization jobs form is populated with all the unique values for the field in the database. Examine the values in the list and decide which normalizing method is best for the existing data. The best practice is to define an alias for a small pool of values and a rule for a large pool of values. The following screenshot shows the pending values for CPU types in Linux servers in a network. The list contains several choices for Intel Xeon CPUs, which might be normalized as Xeon.

1. Navigate to Field Normalization > Normalizations.
2. Open the appropriate normalization record.
3. Click the Normal Values related list.
4. Click New.
5. In the Normal Value form, create normal values for the variants in the Pending Values related list. These are the values the platform uses to replace the variants configured as aliases.

Create aliases
Aliases are the variants of a field value in the instance that will be replaced by the normal value.

The list of potential aliases is the contents of the Pending Values related list. After creating a normal value, assign aliases to this value if the pool of pending values is small. A normalized field can have a combination of aliases and rules.

1. Navigate to Field Normalization > Configurations > Normalizations.
2. Open a normalization record.
3. Click the Normal Values related list.
4. Select one of the values.
5. In the normal value record, click the Aliases related link.
6. Select aliases for this normal value from the available (pending) values that appear in the slushbucket, and then click **OK**.

   The aliases for this normal value now appear in the **Aliases** related list.
Apply aliases

After testing, aliases can be normalized in all new records or in existing records when they are updated. Each time an alias is created for a normal value, a data job is created. The alias is not applied to values in the entire database until its data job is started manually. Run each job separately or run the jobs together to apply all aliases at once.

1. In a normalization record, ensure that the Mode is set to Active.

   Data jobs cannot run in the Test mode.

2. Click the Normal value related list.

3. Select a value from list.

4. In the Normal Value record, select the Data Jobs related list.

   A data job is listed for each alias configured for this normal value.
5. Run the extant data jobs to replace the aliases with the normal value in all existing records in the database.
   1. Select the check box next to a job, and then select **Start** from the Actions menu.
   2. To run all data jobs at once, select all the check boxes, and then select the **Start** action.
   3. Refresh the list to check the progress of the data jobs to ensure that they complete normally.

*Create rules*

The use of rules to normalize a field is intended for large lists of variant field values.

Always test your rules before applying them to all the existing records in the database. Prior to creating the rule, make sure to generate the list of **Pending Values** and create a **normal value** for the field. A normalized field can have a combination of aliases and rules.

The rules in this example are based on the following Pending Values:
Figure 296: Normalization rules 3

1. In a Normalization record, open the Normal Values related list.
2. Open a Normal Value record.
3. Open the Rules related list in the Normal Value form, and then click New.

   The Field Normalization Rule form provides the following fields:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unique name for this rule. For example, this name might be Xeon CPU Type.</td>
</tr>
<tr>
<td>Order</td>
<td>Order in which this rule should be evaluated. The platform parses the rules in the order configured until it finds one that evaluates to true.</td>
</tr>
<tr>
<td>Active</td>
<td>Enables or disables this rule. By default, new rules are active (true)</td>
</tr>
<tr>
<td>Make alias</td>
<td>If this check box is selected, and the rule evaluates to true, the rule makes an alias automatically from a pending value. If this check box is not selected, and the rule evaluates to true, the platform automatically changes the value for the named field in the record to the normal value, but does not create an alias.</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>All pending values for the named field must match the case of the same value in a rule to be normalized.</td>
</tr>
<tr>
<td>Rule</td>
<td>Use the ServiceNow rule builder to construct the rules for normalizing fields. Rules automate the transformation of large numbers of pending field values into aliases.</td>
</tr>
</tbody>
</table>

4. Create a set of rules to incorporate all the possible variants of the pending values.

   For this example, the rules might be:
   - **Intel Xeon**: CPU type matches pattern "Intel*Xeon". This rule normalizes all variants in which Intel precedes Xeon, including Intel Xeon, Intel(R) Xeon(TM) CPU 2.80GHz, and Intel(R) Xeon(TM) CPU 3.00GHz.
   - **Xeon**: CPU type contains Xeon
   - **L3350**: CPU type contains L3350
   - **E3350**: CPU type contains E3350
For every rule that is created, the platform generates a **Rule applier** data job. In the testing mode, the **Start** controls are not available, and the job cannot be run until the mode is changed to **Active**.

5. Test all the rules before making the normalization record **Active**.

**Testing Rules**

**Note:** Users must have the normalization_tester role to create test records.

Field normalization records are created in the **Test** mode by default, enabling administrators to test normalization rules thoroughly before applying them to the existing records in the database. In the testing mode, the **Start** controls are not available for the **Rule applier** data job. The job can be run only when testing is complete and the **Mode** has been changed to **Active**. In the testing mode, only records that have been created or updated by a user with the normalization_tester role are normalized. The normalizer and normalization_tester roles can be combined for a single user or granted separately.

**Coalesce records on a normal value**

Coalescence enables an administrator to redirect references to multiple records containing variants of the same field value to point to a single record, based on a normal value.

An example of this is the Company table that might have multiple variants of a company name, such as Hewlett-Packard, Hewlett-Packard, Inc., Hewlett-Packard Incorporated, HP, and so on. Potentially, thousands of records might reference each of these duplicate company records. Using the variants of the Hewlett-Packard name as aliases, coalescence unifies all these references into a single record that normalizes the **Name** field in the Company record to a normal value such as **HP**.
**Note:** Coalescing normal values changes the record values permanently. If a rollback is performed, records will be returned to the table, but the normalized values will not be rolled back to the original variants.

1. In a normalization record, select the **Coalesce each normal** check box. This applies coalescence to each normal value in the record.
2. Create a normal value for this field, including aliases. An additional field called **Coalesce to** appears on the Normal Value form.
3. Click the magnifier icon in this field and select the specific record from the table (containing this normal value) to which all references should be coalesced.

4. Update the record.
5. **Start** all the **Alias application** data jobs to replace the aliases with the normal value in existing records in the database.

   After these jobs have completed, the **Coalesce to normal** data jobs start automatically for each alias. These jobs redirect all references to normalized records to the selected coalesce record and remove all duplicate records from the database.

*Identify normalized fields*

Normalized fields are marked with an icon that links to different targets, depending on the user’s role.

Users with the normalizer role can click the icon to access the normalization configuration record for that field directly. When a user without the normalizer role clicks the link, a help page appears that provides help on the form. A preference, called Restrict to roles, enables an administrator to define the roles that are permitted to see the icon. If no roles are specified, then the icon is visible to all roles.

---

**Figure 297: Normalization icon**

**Field transformations**

Field transformations enable an administrator to transform raw field input into standardized values that are more meaningful to an organization. Rules control transformations and can be configured for use in queries.

Field transformation standardizes a value in a specific field using rules. An example of this type of standardization might be a name format that removes designations such as Jr. and II from names. Another example is a mode that rounds the amount of computer RAM detected by Discovery to a useful interval.
(4112 MB to 4000 MB). Transform records make up the rules that define how a field transformation is executed. Order values determine the order in which each rule is evaluated. A check box on each transform enables an administrator to determine where processing stops when a rule evaluates to true.

**Note:** The sys_user record that actions the transform process must have its date format set to the default format of "yyyy-MM-dd." Any other date format causes an error during transformation. This problem is only specific to transforming **TO TARGET** fields of type Date/Time. This problem is not an issue if the target field type is of type String or if the field mapping for the date field is changed to the same date format as the transformation process.

**Field transformation scripting**

Scripts that update records or insert records into the database (GlideRecord) are also normalized. For more information on the database, see [GlideRecord](#) on page 3960.

For example, if a script to insert a CI record contains a RAM size of 4112, the script will be transformed to insert the CI with a RAM size of 4000 instead. Scripts that query the database for transformed field values return the normal value (such as 4000) rather than the original (raw) value.

**Create a normalized query for field transformations**

An administrator can configure transformations to apply to queries issued against transformed fields in lists.

For more information on configuration, see [Create a transformation record](#) on page 914.

- In the list, create a filter using the original (raw) value for the transformed field in the query condition.

  ![Filter Example](#)

  **Note:** Normalized queries for field transformations only work for \[is\] or \[is not\] queries.

The filtered list returns records with the transformed value substituted for the raw value. However, the breadcrumbs for the filter display the original query conditions (raw value).

**Field transformations data jobs**

When a Transform is created, a Transform application data job is also created.
This job can only be executed manually and applies the transform to all the appropriate fields in the
database. Data jobs have a roll back feature that allows you to undo a transformation before committing
changes that affect the entire database. Data jobs cannot be started when the Transformation record is in
the **Test** mode. When testing is complete, change the **Mode** setting in the Transformation record to **Active**
to enable the controls for the data job. See *Run a single data job* on page 920 and *Run multiple data jobs*
on page 922 for details about running data jobs.

*Field transformations testing mode*
Transformation is a powerful tool that can cause data issues if used incautiously. Test all transformations
before committing the changes to the CMDB.

Transformation is a powerful tool that can cause data issues if used incautiously. Test all transformations
before committing the changes to the CMDB. All transformation records open in **Test** mode by default,
which blocks administrators from running manual data jobs that apply changes to existing data. Only users
with the normalization_tester role can transform data when a record is in the **Test** mode. This is limited
to new records opened by the tester or records in which the transformed field is updated by the tester. To
apply transformations to the CMDB after testing, change the mode to **Active** in the transformation record
and run the appropriate data job. For information on testing transforms, see *Test a transform* on page
918.

*Transform a field*
This topic lists in order the tasks you need to perform to transform a field.

If you want to also show the original (raw) input in a field on a form after it has been transformed, *create a
raw field*.

1. *Create a transformation record* on page 914.
2. *Select a transform type* on page 915.
3. *Test a transform* on page 918.
4. Run the data job/jobs.
   - *Run a single data job* on page 920.
   - *Run multiple data jobs* on page 922.

Create a transformation record
Creating a transformation record is the first step in transforming a field.

1. Activate the Field Normalization plugin.
2. Navigate to **Field Normalization > Configurations > Transformations**.
3. Click **New**.
4. Create a transformation record.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unique name for this transformation record. This value is for reference only and is not used in any processing.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the ServiceNow table containing the field being transformed. It is important to understand the table hierarchy when setting up a field transform. For example, if you configure transformation for a field in the Computer [cmdb_ci_computer] table, that field will be transformed for all workstation machines, Windows servers, Linux servers, and UNIX servers.</td>
</tr>
</tbody>
</table>
Selecting a transform type

Selecting a transform type is the second step in transforming a field.

1. In the Transformation record, select the **Transforms** Related List.
2. **Click New.**

A selection list of transform types appears, displaying only those transformations appropriate for the field type selected.

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Select the field to transform. The list presented contains only those field types (integer and string) from the table selected that can be transformed.</td>
</tr>
<tr>
<td>Mode</td>
<td>The three available modes are <strong>Off</strong>, <strong>Test</strong>, and <strong>Active</strong>. All transformation records are created in the test mode by default. Do not change the mode until you have thoroughly tested the transformation. When testing is complete, change the mode to <strong>Active</strong>. To disable this transformation, switch the mode to <strong>Off</strong>.</td>
</tr>
<tr>
<td>Normalize query</td>
<td>Select this check box to apply the field value transformed by this record to all queries involving this field. Queries issued with the raw (original) field value will be edited to use the transformation value.</td>
</tr>
<tr>
<td>Raw field</td>
<td>Select the field to use to display the original input (non-normalized) values on a form in which a field value has been normalized. For the selection to appear in the drop-down list, add a custom field to the form for the table selected. For instructions on adding a field for raw data, see <em>Create a raw field</em> on page 938.</td>
</tr>
</tbody>
</table>

5. **Click Submit.**

The **Transforms** and **Data Jobs** Related Lists appear on the form.
Choose the type of transform you want to create:

Text  *Transforms that modify the text of the value.*

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Case</td>
<td>Adjusts the case of characters in the value</td>
</tr>
<tr>
<td>Constant</td>
<td>Supply a constant value</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete specified sequence of characters from the value</td>
</tr>
<tr>
<td>Insert</td>
<td>Insert a fixed character sequence into the value</td>
</tr>
<tr>
<td>Left</td>
<td>Keep or delete the leftmost 'n' characters</td>
</tr>
<tr>
<td>Prefix</td>
<td>Adds characters to the beginning of the value</td>
</tr>
<tr>
<td>Replace</td>
<td>Replaces occurrences of one string with another</td>
</tr>
<tr>
<td>Right</td>
<td>Keep or delete the rightmost 'n' characters</td>
</tr>
<tr>
<td>Substring</td>
<td>Keep or delete characters from a specified sub-sequence of characters in the value</td>
</tr>
<tr>
<td>Suffix</td>
<td>Append characters to the end of the value</td>
</tr>
<tr>
<td>Trim</td>
<td>Remove any leading or trailing spaces or tabs</td>
</tr>
</tbody>
</table>

Figure 298: Transform types

3. Select a transform type and provide the appropriate parameters.
4. Select an **Order** number for this transform.

**Note:** The conditions for the transforms are executed according to the order numbers assigned.

5. Select the **Final** check box to stop processing with this transform if the condition evaluates to true.
6. Select the **Case sensitive** check box to force case sensitivity in the condition statement.
7. To create a condition that uses regular expressions or pattern matching, select the correct operator.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>regular expressions</td>
<td>matches regex</td>
</tr>
<tr>
<td>pattern matching</td>
<td>matches pattern</td>
</tr>
</tbody>
</table>

The following transform example replaces the INC at the beginning of an incident number with the string ENG if the assignment group is ITSM Engineering.
Figure 299: Transformation record

8. Click **Submit**.

The new Transform appears in the Related List of the Transformation record.
When the Transform is created, a Transformation application data job is also created. This data job applies this transform to appropriate records in the entire database and should not be run until testing is complete.

Test a transform
Testing a transform is the third step in transforming a field.
Note: Users must have the normalization_tester role to create test records.

New transformation records open in the Test mode by default, enabling administrators to test transforms thoroughly before applying them to the existing records in the database. In the test mode, the Start controls are not available for the Transform application data job. There are two methods, listed below, for testing transforms before committing the transformations to existing data.

- Create or update records.
  In the test mode, only records that have been created or updated by a user with the normalization_tester role are transformed. Grant the normalizer and normalization_tester roles to the same user or grant them to separate users.
- Test transforms utility.
  This feature enables a normalization tester to transform field values on the fly without opening or updating records. This utility tests all the transforms configured for this field.
  a) Open a Transformation record.
  b) Click the Test transforms Related Link.

![Test transforms](image)

A dialog box appears for testing field values.

c) Enter a value to transform in the Raw data field.
d) Click **OK**.

The platform transforms the raw value in the **Transformed data** field.

e) Enter new raw data to test other transforms.

f) Click **Cancel** to end the test.

g) When testing is complete, change the **Mode** to **Active** and run the data job.

---

Run a single data job

Running the data job/jobs is the fourth step in transforming a field. This topic explains how to run a single data job.

1. Start the Transform application data job to apply the transform to all the appropriate records in the database.

2. Open the Transformation record and switch the **Mode** to **Active**.

3. Select the **Data Jobs** Related List.

4. Run the data job using one of the following methods.

   - In the list of data jobs, select the check box of the job you want to run, and then select **Start** from the **Actions** menu. See #unique_1077/unique_1077_Connect_42_f_ModeSetToActive on page 921.

   - Click the link in the Created column to open the data job, and then click the **Start** Related Link. See #unique_1077/unique_1077_Connect_42_f_StartRelatedLink on page 921.
The State of the jobs turns to **Queued** as it runs and to **Completed** when it finishes successfully.
Run multiple data jobs

Running the data job/jobs is the fourth step in transforming a field. This topic explains how to run data jobs for multiple field transformations.

1. Start the Transform application data job to apply the transform to all the appropriate records in the database.
2. Navigate to Field Normalization > Data Jobs > All.
3. Select the check boxes for the jobs you want to start.

   **Note:** These jobs must have the Mode set to Active.

4. Select Start from the Actions menu.

   **Note:** The platform only runs data jobs from Active transformations. The Action menu indicates the number of Active data jobs that can run. For example, the menu might display Start (3 of 4).

**Transformed field identification**

Transformed fields in records are marked with an icon that links to different targets, depending on the user's role.

Users with the normalizer role can click the icon for direct access to the transformation configuration record for that field. When a user without the normalizer role clicks the link, a Wiki page appears that provides help on the form. A preference, called Restrict to roles, enables an administrator to define the roles that are permitted to see the icon. If no roles are specified, then the icon is visible to all roles.
Transform definitions define the transformation actions available for a given field type.

Users select a definition when they transform a field, and then provide the definition with the specific parameters that are applied to the transformation. For example, a definition can round up an integer or insert a value at a defined position in a string. The system provides a number of definitions that are designed to meet the needs of most organizations. If an administrator needs a transformation action that is not available in the out-of-box configuration, he can create a new definition. Transform definitions can be associated with existing Transform Categories or to new categories.

Create a transform definition
The following example describes the procedure for creating a new transform definition. In this example, we create a definition that transforms a number field to an odd or even integer. The transform category is Numeric and the normalization field type is Integer.

1. Navigate to Field Normalization > Administration > Transform Definitions.
2. Click New in the record list.
3. Enter a name for this definition.

   Note: In this example, we enter Odd/Even.

4. Enter a brief description of the action, such as, Transforms an integer to an odd or even value.
This information appears in the definition choice list when a user selects a new transform.

5. Right-click in the header bar and select Save in the context menu.

Two Related Lists appear in the form.

- **Transform Categories:** Click **Edit** and select **Numeric** as the category to which this definition belongs. Currently, field transformation supports two categories: **Numeric** and **Text**. The **Integer** normalization field type is already associated with this category.
- **Transform Variables:** Define any variables required by this transform definition to perform an action on a field value. Variables are not necessary if a script can perform the action alone.

Create a transform variable for a transform definition

Transform variables enable an administrator to apply the same definition to different fields in different ways.

Transform variables contain values used by a script to perform a field transformation. Scripts and variables can be created in either order, but the script must use the transform variables. Transform variables are populated with values when a user configures a transform type.

1. In the Transform Definition record, click **New** in the Transform Variables Related List.
2. Complete the form.

Important considerations for completing a form:
• The Column name is an entry in the fn_transform_var table for this variable. This becomes the variable in the script, in the form of variables.<variable name>. For this example, we enter odd_even.
• The value in the Label field appears as the variable field label in the Transform form. In our example we enter Odd/Even.
• The field Type defines the field type of the variable value. Because the values for the variables used are "even" and "odd", this is a type of string.
• The Order of the variables controls the order in which they are displayed in lists and records.
• This variable has a choice list with two options: Even and Odd. We select Dropdown without - None as our format for the list in the Choice field and define a Default value of even when the list is displayed.
• Create a Hint that becomes a tooltip for the variable in the Transform record.

3. Right-click in the header bar and select Save from the context menu.

The Variables Choice List Related List appears.
4. Click **New** in the Variables Choice List and define the list options.

5. Create records for **Even** and **Odd**.

**Note:** The **Element** value is the same as the **Column** name in both selections for the choice list.
Create a script for a transform definition

Create the script at any time during the configuration of a definition.

The script can perform a transform action without using a variable, but the action of the definition will be the same for all fields. Variables create more flexibility for the definition, enabling an administrator to use the same definition in different ways in different places. If a variable is defined, the script must reference the variable using the correct format.

There are three arguments in the script:
• Variables: Contains the variables using the format variables.<variable name>.
• Value: Contains the un-transformed value
• Parameters: Special objects that set debug messages.

1. Open the **Odd/Even** record in the Transform Definitions module.
2. Enter the following script to pass values with the odd_even variable.

```javascript
function(variables, value, parameters) {
  var odd = ('odd' == variables.odd_even);
  var val = value - 0;
  var val_odd = ((val & 1) == 1);
  if (odd != val_odd)
    val++;
  return '' + val;
}
```
Notice that the script references the variable in the form variables.odd_even.

3. Update the record to complete the configuration.

The Odd/Even transform definition is now ready to use in a field transformation.

Definitions included with field transformation
All position parameters (Starting position, Ending position, etc.) have three modes that apply to all the transform types that use this variable.

Table 228: Position parameters

| Positive positions | If the position is expressed as a positive integer, the platform calculates the starting position beginning from the left side of the field value. For example, in the string ABCDE, a position of 3 places the starting point of the action after C. |
Negative positions

If the position is expressed as a negative integer, the platform calculates the position beginning from the right side of the field value. For example, in the string ABCDE, a position of -3 places the starting point of the action before C.

Regex

If the position value starts with /regex/, everything after that is a regular expression that is used to calculate the starting position. For example, in the string ABCDE, a position of /regex/B.*D places the starting point of the action after C (B and all the characters between B and D).

<table>
<thead>
<tr>
<th>Transform Type</th>
<th>Category</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Change case    | Text     | Changes the case of the characters in the field value. | Mode: Select one of the following modes:  
  • Upper: Converts the value to all upper case characters  
  • Lower: Converts the value to all lower case characters  
  • Proper: Converts the value to title case, with the first character in each string in upper case, and the remaining characters of the string in lower case.  
  • Formal: Converts the value to a string in which only the first letter of the first word is in upper case. |
<p>| Constant       | Text Numeric | Converts the value in this field to a constant. | Constant: The constant with which to replace the value in this field. |</p>
<table>
<thead>
<tr>
<th>Transform Type</th>
<th>Category</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Text</td>
<td>Delete a specified sequence of characters from a field value.</td>
<td>- Starting position: Specifies the first character in a sequence of characters to delete from a string. See the discussion of position modes at the beginning of this section for details. - Ending position: Specifies the final character in a sequence of characters to delete from a string. See the discussion of position modes at the beginning of this section for details.</td>
</tr>
<tr>
<td>Insert</td>
<td>Text</td>
<td>Insert a fixed character sequence into a field value.</td>
<td>- Position: The character position at which to insert the new value. See the discussion of position modes at the beginning of this section for details. - Insert: The value to insert into this field.</td>
</tr>
<tr>
<td>Left</td>
<td>Text</td>
<td>Deletes or keeps a specified number of characters from the left side of this field value.</td>
<td>- Position: Specifies the number of characters to keep or delete from the left side of the value. See the discussion of position modes at the beginning of this section for details. - Mode: Select the mode for this transform: Keep or Delete.</td>
</tr>
<tr>
<td>Prefix</td>
<td>Text</td>
<td>Adds characters to the beginning of a field value.</td>
<td>Prefix: Defines the characters to add to the beginning of the transformed field value.</td>
</tr>
<tr>
<td>Transform Type</td>
<td>Category</td>
<td>Description</td>
<td>Parameters</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| Replace        | Text     | Replaces occurrences of one string with another string. The special characters backslash (\) and dollar sign ($) in the replacement string can cause the transform to be different than if the replacement string were being treated as a literal replacement string. Use a regular expression to replace a string or parts of a string. | • Find: Enter the string or regular expression to replace.  
• Replace with: Enter the replacement string. |
| Right          | Text     | Retains or deletes a specified number of characters from the right side of a field value. | • Position: The number of characters to delete or keep from the right side of this transformed field. See the discussion of position modes at the beginning of this section for details.  
• Mode: Select the mode for this transform: Keep or Delete. |
<table>
<thead>
<tr>
<th>Transform Type</th>
<th>Category</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Round          | Numeric  | Rounds integers to a configured rounding interval using specific criteria. The interval must be appropriate to the value being transformed, such as an interval of 12 for a value expressed in dozens or 0.01 for decimal values expressed in hundredths. | - Interval: Select the rounding interval that is appropriate to the units of the field value. For example, an interval of 256 is appropriate for expressing RAM values in megabytes, but does not work for Disk space expressed in gigabytes. The rounding interval for the examples below is 1  
- Mode: Criteria for applying the rounding interval.  
  - Half up: Always round up a value that is exactly half way between two intervals. For example, 3.5 is always rounded up to 4, and -3.5 is always rounded up to -3.  
  - Half down: Always round down a value that is exactly half way between two intervals. For example, 3.5 is always rounded down to 3, and -3.5 is always rounded down to -4.  
  - Half away from zero: Always round an integer that is half way between the specified interval away from zero. For example, 3.5 is always rounded to 4, and -3.5 is always rounded to -4.  
  - Half toward zero: Always round an integer that is half way between the specified interval toward zero. For example, 3.5 is always rounded to 3, and -3.5 is always rounded to -3.  
  - Half to even: Always round an integer that is half way between the specified interval to the nearest interval whose least significant digit is even. For example, 3.5 is always rounded to 4, and 4.5 is always rounded to 4.  
  - Half to odd: Always round an integer that is half way between the specified interval to the nearest interval whose least significant digit is odd. For example, 3.5 is always rounded to 3, and 4.5 is always rounded to 5.  
  - Up: Always round an integer up by the specified rounding interval. For example, 3.4 is always rounded to 4 by a rounding interval of 1.0.  
  - Down: Always round an integer down by the specified rounding interval. For example, 4.6 is always rounded to 4 by a rounding interval of 1.0.  
  - Away from zero: Always round an integer away from zero by the specified rounding interval. For example, 3.3 is always rounded to 4, and -3.3 is always rounded to -4 by a rounding interval of 1.0.  
  - Toward zero: Always round an integer toward zero by the specified rounding interval. For example, 3.3 is always rounded to 3, and -3.3 is always rounded to -3 by a rounding interval of 1.0. |
<table>
<thead>
<tr>
<th>Transform Type</th>
<th>Category</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Substring     | Text         | Keep or delete characters from a specified sub-sequence of characters in the field value. | • Starting position: Specifies the first character in a sub-sequence of characters within the value. See the discussion of position modes at the beginning of this section for details.  
• Ending position: Specifies the final character in a sub-sequence of characters within the value. See the discussion of position modes at the beginning of this section for details.  
• Mode: Select whether to Delete the sub-sequence selected or Keep only those characters defined. |
| Suffix        | Text         | Appends characters to the end of a field value.                             | Suffix: Defines the suffix to add to the end of the field value.           |
| Trim          | Text Numeric | Removes blank spaces from the field value.                                 | No parameters                                                             |

**Transform categories**

Transform categories are used to group the transform definitions together appropriately to present to users when creating new field transforms.

You can create new categories for existing definitions or change the default associations of categories to definitions. Transform categories are associated with normalization field types to present the correct definition option list for the field being transformed. The base system provides two transform categories: Text and Numeric.

- **Text**: Definitions in the Text category transform string type field values. Included in the Text category are:
  - Left
  - Right
  - Constant
  - Trim
  - Prefix
  - Suffix
  - Change Case
- Delete
- Insert
- Substring
- Replace

- Numeric: Definitions in the Numeric category arithmetically manipulate integer type field values. Included in the Numeric category are:
  - Limit
  - Round
  - Constant

Create a transform category

1. Navigate to Field Normalization > Administration > Transform Categories.
2. Click New in the Transform Categories Related List.
3. Enter the Name of this category and a description.
4. Select an Order for this category and save the record.

The order determines the display order of categories in lists and forms. Two Related lists appear:

- **Field Types**: Click Edit to select an existing field type for this category or click New to create a new field type. The normalization field types provided are:
  - Decimal
  - Float
  - Integer
  - Numeric
  - String
  - URL

- **Transform Definitions**: Click Edit to select the transform definitions that are included in this category.

The completed category form looks like this:
Regular expressions and patterns in field normalization rules
Field Transformation definitions support the use of regular expressions (referred to in the platform as *regex*) and pattern matching for determining the position of characters in a string.

After identifying the target characters, field transformation can replace or delete the identified characters or insert other characters at that position.

**Regex**

Regular expressions can be used in transform parameters and in condition statements to determine which characters in a field value are transformed.
• Transform parameters: Regular expressions used as parameters to locate characters in transformed field values must begin with \regex/. Everything after that is a regular expression that is used to calculate character position. For details, see Transform definitions on page 923.

• Conditions: Condition statements are used to configure the transforms that select the characters to change in a field value. When using a regular expression in a condition statement, make sure to select the matches regex operator.

Example

The computer names in an organization's Windows network are expressed as domain\machine name, such as development\devlab01. The network administrator wants to simplify these names by removing the domain name and backslash. He creates a transformation record for the Computer [cmdb_ci_computer] table and selects the Name field to transform.

![Figure 303: Transformation regex example](image)

The network contains several domains, and each domain contains numerous computers. The only character common to each name is the backslash. To delete the domain name, the administrator decides to use a regular expression to replace the entire raw value in the field with the characters that appear after the backslash (the actual machine name). He creates a new Transform using Replace as the Transform Type and enters the following values:

• Find:\regex/.\(\.*\)
• Replace with:$1
Figure 304: Transformation regex example

The regular expression `.*\(.*\)` represents the entire raw value in the **Name** field - in this example `development\devlab01`. The first part of the expression, `.*`, represents everything before the backslash (the **development** domain name). The backslash by itself is the escape character in regular expressions and requires special syntax to retain its function in the computer name. The administrator must *escape* it by using another backslash (`\` means `\`). The part of the expression after the backslash, `(.*`, represents the computer name (`devlab01`) and is grouped within parentheses for reference. The value in the **Replace with** field, `$1`, references this group and replaces the entire raw value of the field with the contents of the group, `devlab01`.

The administrator clicks **Test transforms** in the transformation record and enters `development \devlab01` in the **Raw data** field. He then clicks **OK** to apply the transform to the test value. The transform replaces `development\devlab01` with `devlab01`.

Figure 305: Transformation Regex 3

When the transforms for this field are tested successfully, the administrator changes the **Mode** in the transformation record to **Active** and runs the Transformation application data job to apply this transformation to existing records in the database.

**Pattern matching**

Pattern matching in Field Normalization uses special characters differently from regular expressions to create patterns that the platform recognizes when transforming field values.
Pattern matching can be used only in condition statements. When using pattern matching characters in a condition statement, make sure to select the matches pattern operator.

Use the following special characters to create patterns for searches.

- The asterisk in a search string (*) matches any number (including zero) of any character.
- The question mark (?) in a search string matches one of any character.
- Everything else in a search string matches itself.

Examples

- the story matches the story but not that story.
- *story matches the story and that story, but not that story is the best.
- st?ry matches story and stxry, but not my story or stairy.
- *b?gus* matches bogus, my bogus story, and His bagus machine, but not my bgus story or my baigus story.

Create a raw field

A raw field is a custom field created by an administrator to show the original (raw) input in a field on a form after it has been normalized or transformed.

1. In the form containing the field that is being normalized or transformed, right-click in the header bar.
2. Select Configure > Form Layout.
3. Complete the Create new field form at the bottom of the page, and then click Add:
   - Name: Type the field label. In this case, use Raw + <field label>.
   - Type: Select a data type from the list for this field.
   - Field Length: Select the character limit for this field. The default is 40.
4. Move the new field adjacent to the normalized field using the direction arrows in the slushbucket.
5. Click Save.

Data policy

Data policies enable you to enforce data consistency by setting mandatory and read-only states for fields.

Data policies are similar to UI policies, but UI policies only apply to data entered on a form through the standard browser. Data policies can apply rules to all data entered into the system, including data brought in through import sets or web services and data entered through the mobile UI.

For example, suppose that you are configuring a web service that allows users from outside the platform to update problems on the ServiceNow instance. Since these problems are not updated through the instance UI, they are not subject to the UI policies on the problem form. To ensure that the Close notes field is completed before a problem is marked Closed/Resolved, you can create a data policy that applies to server-side imports. Data that does not comply with this data policy produces an error. You can also apply the policy on the browser by selecting the Use as UI Policy on client check box in the data policy record.

Since UI policies can also manage the visibility of fields on a form, you may want to augment UI policies with data policies rather than replace them.

By default, data policies are applied to all GlideRecord operations including those used in Scripted REST APIs, and the REST Table API. You can opt out of applying the data policy to:

- Target records of web services
• Import sets
• Client-side UI policies

The admin role is required to edit data policies.

**Note:** Defining a data policy enforces the policy when a record is submitted from the UI. This behavior cannot be changed.

---

**Installed with data policy**

What is installed with Data Policy.

**Tables**

- **Data Policy** [sys_data_policy2]
- **Data Policy Rule** [sys_data_policy_rule]

**Roles**

The following role is available with data policy:

- **data_policy_admin**: Allows you to grant access for maintaining data policy to specific users. You may have to grant this role access to the System Policy application.

**Data Policy Module**

The **System Policy > Data Policies** module displays a list of all data policies and where they apply.

**Figure 306: Data policy list 2**

![Data policy list 2](image_url)

**Data policy fields**

These fields appear on the Data Policy form and related forms.
### Table 230: Data policy fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>The table to which this policy applies.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The list shows only tables and database views that are in the same scope as the data policy.</td>
</tr>
<tr>
<td>Application</td>
<td>Application that contains this data policy.</td>
</tr>
<tr>
<td>Inherit</td>
<td>If selected, applies this data policy to tables that extend the specified table. For example, incident, problem, and change tables all extend the task table, therefore selecting Inherit on a data policy defined for task would apply the data policy to them as well.</td>
</tr>
<tr>
<td>Reverse if false</td>
<td>If selected, the data policy action is reversed when the conditions evaluate to false. For example, when the conditions are true, then actions are taken and when they change to false, the actions are reversed.</td>
</tr>
<tr>
<td>Active</td>
<td>If selected, the data policy is used.</td>
</tr>
<tr>
<td>Short description</td>
<td>A short description that identifies the policy.</td>
</tr>
<tr>
<td>Description</td>
<td>A detailed description of the policy.</td>
</tr>
<tr>
<td>Apply to import sets</td>
<td>If selected, the data policy applies to data brought into the system from import sets. This option also applies to web service import sets.</td>
</tr>
<tr>
<td>Apply to SOAP</td>
<td>If selected, the data policy applies to data brought into the system from a SOAP web service. Scripted SOAP web services are not affected. This field does not affect data policy interaction with REST web services.</td>
</tr>
<tr>
<td>Use as UI Policy on client</td>
<td>If selected, enforces the data policy on the UI using the UI policy engine.</td>
</tr>
</tbody>
</table>

### Table 231: Data policy rule fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>The table on which the data policy action applies.</td>
</tr>
<tr>
<td>Field name</td>
<td>The field from the specified table to which the data policy will apply.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Read Only</td>
<td>How the data policy affects the read only state of the field. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• Leave alone</td>
</tr>
<tr>
<td></td>
<td>• True</td>
</tr>
<tr>
<td></td>
<td>• False</td>
</tr>
<tr>
<td>Mandatory</td>
<td>How the data policy affects the mandatory state of the field. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• Leave alone</td>
</tr>
<tr>
<td></td>
<td>• True</td>
</tr>
<tr>
<td></td>
<td>• False</td>
</tr>
</tbody>
</table>

**Note:** For tables that are in a different scope than the data policy record, you cannot make a field mandatory.

Create a data policy

You can create a new data policy to define data rules for a table.

Create data policies to enforce consistency. You can create data policies only for tables and database views that are in the same scope as the data policy and for other tables that have at least one field in the same scope as the data policy. For tables that are in a different scope than the data policy record, you can create data policy rules only for fields in the same scope as the data policy and you cannot make a field mandatory.

1. Navigate to **System Policy > Rules > Data Policies**. Alternatively, from any list header or form header, right-click in the header bar and select **Configure > Data Policies**.
2. Click **New**.
3. Select any options for the data policy.
4. Create the condition that must exist for the platform to apply this policy.
   For example, your conditions might include **[Problem state] [is] [Closed/Resolved]**
5. Right-click the header and select **Save**.
   The **Data Policy Rules** related list appears.
6. Click **New** in the related list and create the record that identifies the field and the policy to apply.

![Data Policy Rules](image.png)

It is possible to have multiple rules on a single field, but it is not recommended.

7. Click **Submit**.
8. Optional: Add more rules by repeating steps 6 and 7.

Convert a UI policy to a data policy

An administrator can convert a UI policy to a data policy.

Existing UI policies can be converted to data policies. This enables you to take a policy that currently applies only to records created or updated in the browser and extend the policy to apply to import sets and data imported by SOAP web services, or to apply by default.

For a UI policy to be eligible for conversion to a data policy, the following three conditions on the UI Policy form must be met:
• The **Run scripts** option must not be selected
• The **Global** option must be selected
• None of the UI policy actions can have **Visible** set to **True** or set to **False** (it must be set to **Leave Alone**)

Note that converting a UI policy to a data policy deactivates the UI policy. To retain the policy in the UI, ensure that the **Use as UI Policy on client** checkbox is selected on the data policy record.

1. Navigate to **System UI > UI Policies** and click an existing UI policy.
2. Under **Related Links**, click **Convert this to Data Policy**.

A new data policy record is created.

3. Edit the fields on the data policy record as necessary.

**Convert a data policy to a UI policy**

Converting a data policy to a UI policy is useful if a data policy already exists, but only needs to apply to records created or updated in the browser.
Converting deactivates the data policy - the new UI policy is applied only at the UI layer only and not to import sets or data imported from SOAP web services.

**Note:** An alternative to converting from a data policy to a UI policy is to select the **Use as UI Policy on client** checkbox on the data policy record. This field extends the data policy to the UI. The main difference between converting and using the **Use as UI Policy on client** checkbox is that converting provides the **Visible** field on the UI policy record. Use the **Visible** field to select how the UI policy affects the visible state of the field.

1. Navigate to **System Policy > Rules > Data Policies** and click an existing data policy.
2. Under **Related Links**, click **Convert this to UI Policy**.

A new UI policy record appears.

3. Edit the fields on the UI policy record as necessary. For details about the fields, see *Create a UI Policy*.

**Data policy debugging**

Debug messages can help administrators identify and resolve data policy problems.

Debug messages can help you identify and resolve data policy problems. To view data policy debugging messages at the bottom of the screen, navigate to **System Diagnostics > Session Debug > Debug Data Policies**.
In the example, a data policy is in place to prevent the short description on an incident from being changed when the incident state is set to Open. A user edited the short description while the incident was open and tried to save the changes, but the data policy was enforced.

![Debug Output]

Figure 307: Data policy debug messages

Data lookup and record matching support

The data lookup and record matching feature enables administrators to define rules that automatically set one or more field values when certain conditions are met.

Data lookup rules allow administrators to specify the conditions and fields where they want data lookups to occur. For example, on Incident forms, there are priority lookup rules for the sample data that automatically set the incident **Priority** based on the incident **Impact** and **Urgency** values.

Create custom data lookups

Creating new data lookups involves several steps.

1. **Create a custom data lookup table.**
2. **Add data lookup values to the data lookup table.**
3. **Create a Data Lookup Definition record.**
4. **[Optional] Create a data lookup module.**

**Create a custom data lookup table**
Create a custom table to store lookup data.

The custom table must extend the Data Lookup Matcher Rules [dl_matcher] table. For example, this custom lookup table stores information about VIP callers and incident assignments.

<table>
<thead>
<tr>
<th>Field</th>
<th>Sample value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>VIP Caller Lookup</td>
</tr>
<tr>
<td>Table name</td>
<td>u_vip_caller_lookup</td>
</tr>
<tr>
<td>Extends base table</td>
<td>dl_matcher</td>
</tr>
<tr>
<td>Create new module</td>
<td>True</td>
</tr>
<tr>
<td>Add module to menu</td>
<td>System Policy</td>
</tr>
</tbody>
</table>

**Add a data lookup value to the data lookup table**
The columns of a data lookup table contain both matcher and setter field data.
Each data lookup is a query that searches for a row containing values that match the matcher fields. The data lookup then returns the value listed in the setter fields. For example, this Priority Data Lookup [dl_u_priority] table lists the combinations of impact and urgency (matcher fields) that produce a particular priority value (setter field).

Table 233: Lookup table

<table>
<thead>
<tr>
<th>Matcher fields</th>
<th>Setter field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impact</td>
<td>Urgency</td>
</tr>
<tr>
<td>1 - High</td>
<td>1 - High</td>
</tr>
<tr>
<td>1 - High</td>
<td>2 - Medium</td>
</tr>
<tr>
<td>1 - High</td>
<td>3 - Low</td>
</tr>
<tr>
<td>2 - Medium</td>
<td>1 - High</td>
</tr>
<tr>
<td>2 - Medium</td>
<td>2 - Medium</td>
</tr>
<tr>
<td>2 - Medium</td>
<td>3 - Low</td>
</tr>
<tr>
<td>3 - Low</td>
<td>1 - High</td>
</tr>
<tr>
<td>3 - Low</td>
<td>2 - Medium</td>
</tr>
<tr>
<td>3 - Low</td>
<td>3 - Low</td>
</tr>
</tbody>
</table>

1. In the navigation filter, enter the name of the new custom lookup table. For example, enter u_vip_caller_lookup.list.
2. Configure the list and create new fields, as appropriate. For this example, create the following new fields:

Table 234: New fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Field length or Table to reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caller</td>
<td>Reference</td>
<td>User [sys_user]</td>
</tr>
<tr>
<td>Priority</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>Assignment Group</td>
<td>Reference</td>
<td>Group [sys_user_group]</td>
</tr>
</tbody>
</table>

3. From the table list, click New and enter appropriate matcher and setter field values. For example:

Table 235: New matcher and setter field values

<table>
<thead>
<tr>
<th>Matcher field</th>
<th>Setter fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caller</td>
<td>Priority</td>
</tr>
<tr>
<td></td>
<td>Assignment Group</td>
</tr>
<tr>
<td>Beth Anglin</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>VIP Issues</td>
</tr>
<tr>
<td>Fred Luddy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>VIP Issues</td>
</tr>
</tbody>
</table>
Create a data lookup definition record
How to create a data lookup definition record.

2. Click New.
3. Click Data Lookup Rule.
4. Fill in the fields on the form, as appropriate (see table).
5. Right-click the form header and click Save.
6. From the Matcher Field Definitions related list, click New.
7. Fill in the Matcher Field Definitions fields.
8. Click Submit.
9. From the Setter Field Definitions related list, click New.
10. Fill in the Setter Field Definitions fields.
11. Click Submit.
12. Click Update.

For example, the following data lookup definition assigns incidents to the VIP Issues group based on the Caller field. In addition, the incidents are set to critical or high priority based on the caller.

Note: Each row in a data lookup table must be unique.
Data lookup definitions fields

Data lookup definitions fields.

Table 236: Data lookup definitions fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name to identify the definition record.</td>
</tr>
</tbody>
</table>
Geneva    ServiceNow    ServiceNow Platform

Field Description
Source Table Select the table containing the fields you want to automatically update with lookup values. Data Lookup Definitions are not inherited by extension tables. For example, a Data Lookup Definition on the Task table cannot match values on the Incident incident table.

Matcher Table Select the table containing the lookup values. This table should always start with a u_ prefix.

Active Select this check box to run this data lookup rule. Clear the check box to ignore this data lookup rule.

Run on form change Select this check box to automatically look up values whenever a user or onChange client script changes a field value on a source table form.

Note: This does not include changes automatically made by other data lookup rules, such as the Priority Lookup Rules.

Run on insert Select this check box to automatically look up values whenever a user creates a new record.

Run on update Select this check box to automatically look up values whenever a user saves or updates a record.

Matcher field definitions fields

The matcher field definitions determine when a data lookup occurs.

A data lookup only occurs on fields with matcher field definitions. The data lookup uses the values of the source table fields to look up one or more values from the matcher table. Note that data lookup does not work with Journal type fields.

Table 237: Matcher field definitions fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Lookup</td>
<td>Displays the name of the parent data lookup definition record.</td>
</tr>
<tr>
<td>Source table field</td>
<td>Select the field from the source table that contains the data to match.</td>
</tr>
<tr>
<td>Matcher table field</td>
<td>Select the field from the matcher table that contains the data to match.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved. 949
Setter field definitions fields

The setter field definitions determine what fields the data lookup changes when the matching conditions are true.

Table 238: Setter field definitions fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Lookup</td>
<td>Displays the name of the parent data lookup definitions record.</td>
</tr>
<tr>
<td>Source table field</td>
<td>Select the field from the source table that the data lookup updates.</td>
</tr>
<tr>
<td>Matcher table field</td>
<td>Select the field from the matcher table that provides the new value for the update.</td>
</tr>
<tr>
<td>Always replace</td>
<td>Select this check box to replace any existing value with a value from the data lookup. Clear this check box to ignore the update if the field has an existing value.</td>
</tr>
</tbody>
</table>

Create a data lookup module

How to create a data lookup module.

 Optionally, you can create a new module for the data lookup table you created in Step 2 to simplify access to the matcher table records. The module must have these properties:

Table 239: Module properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Required value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select the data lookup table you created in Step 2. For example u_vip_caller_lookup.</td>
</tr>
</tbody>
</table>
## Troubleshooting data lookup

If the custom data lookup definition rules are not behaving as expected, check for certain conditions.

- Verify that the data lookup definition is set to run on the appropriate events.
- Verify that the matcher field is not read-only. Since users cannot change read-only fields, user interactions cannot trigger an on form change event for read-only fields.
- Verify a client script is not changing a field value. Client scripts can trigger Run on form change events even on read-only fields.
- Verify that the data in the matcher table is correct.
- If the lookup requires an exact match, verify that there is a matcher table row for each possible combination (including blank values). The lookup fails if cannot find a matching value.
- Verify that you have not created a recursive rule, such as:
  
  If Field A = 1, then Field B =2. If Field B = 2, then Field A = 2

## Activate data lookup and record matching

The Data Lookup and Record Matching Support plugin is automatically active for new instances.

If you upgrade from a previous version, you can manually activate the Data Lookup and Record Matching Support plugin if it is not already active. See [Activate a plugin](#) on page 1233.

---

**Note:** Activating the Data Lookup and Record Matching Support plugin replaces the calculatePriority business rule with a priority data lookup definition, but does not transfer any custom logic. If you manually activate the plugin, you must recreate any custom business logic that uses the priority lookup rules.

---

## Suggestion fields

A suggestion field allows users to select predefined text in addition to entering text.

Suggested text can be configured for any text or journal field.

## Use a suggestion field

A suggestion field allows users to select predefined text in addition to entering text.

Role required: none

Any text or journal field may offer suggested text, depending on setup.

1. Click the light bulb icon next to a suggestion field.
2. Click the suggested text to enter. For long text fields, select a label to view suggested text and then click **Select** to enter the response.
Add a suggestion field

You can add a suggestion field to a form.
Role required: personalize_dictionary

1. Open the form to which you are adding a suggestion field.
2. Add a new field or locate an existing field.
3. Right-click the field label and select **Configure Dictionary**.
4. In the **Choice** field, select **Suggestion**.
5. Click **Update**.

Configure suggested text for string fields
Suggestions are configured differently depending on field type.

Role required: personalize_choices

You can configure the suggested text options for string fields.

1. Right-click the field label and select **Configure Choices**.
2. Using the slushbucket, select options and the order in which you want them to appear.
   - To create a new option, enter the suggested text in the **Enter new item field** and click **Add**.
   - The **Apply to Table** field is available when the current table extends another table (for example, Incident extends Task). This field allows suggested text options to be configured for all tables that extend the parent or for only the current (child) table.
3. Click **Save**.
4. To edit existing options, right-click the field label and select **Show Choice List**.
Configure suggested text for journal fields
Suggestions are configured differently depending on field type.
Role required: personalize_dictionary

You can configure the suggested text options for journal fields (field types that begin with Journal).

1. Right-click the field label and select **Configure Responses**.
2. Click **New**.
3. Enter a label, or brief description, for the option.
4. Enter the complete text in **Response text**.
5. Click **Save**.
Currency administration

Currency fields provide features for handling display and calculations of currency values.

A currency field holds a value, a currency code, and a reference currency value.

- The currency code is a three-letter ISO code that identifies the currency in which the value is specified.
- The reference currency value is a number representing the currency value in the reference currency. The reference currency value is calculated by a rate conversion when the currency value is saved.

A price field is similar to a currency field, but with special features for conversion and display. To learn more, see Price fields on page 957.

Locale settings

There are two locale settings, system and user. The system locale determines the reference currency, and the user locale determines the session currency.

The system locale is set using the glide.system.locale property. The value is of the format Language.Country where the language is an ISO 639 language code and the country is an ISO 3166 language code. Internally, this value is used as specified by Java. The system locale setting should be in the Java supported locales list. The system locale should be set once on a fresh zboot because reference currency values in currency fields are assumed to be in the currency implied by the system locale.
**Note:** Do not change the system locale after currency values have been entered into the instance. When you change the system locale, the reference currency values are not adjusted, that is, there is no rate conversion. This persistence results in invalid aggregations and filtering.

The user locale is determined by the following, in order of consideration.

- User record in which both country and language are specified.
- System locale set using the property glide.system.locale.
- Browser locale.

### Session and reference currency

The system uses two kinds of currency, session and reference.

The session currency is defined for the user by the user's locale or single-currency mode. The reference currency is determined by the system locale. The reference currency is a standard used across the entire instance. Each time a value is entered in a currency or price field, the system stores three pieces of information:

- The value as entered, in the user's locale.
- The currency code, in the user's locale.
- The value converted to the reference currency using the current exchange rate.

**Note:** In multiple-currency mode, the currency code saved in currency field may not be the same as the session currency code. For example, the session currency could be the Euro and the number entered could be the Japanese Yen.

### Session currency

When users view a currency value, they can see the value as entered or in the session-currency format. The format contains:

- The currency symbol
- The value converted to the session currency and shown in a localized number format.

The user's locale determines the session currency format.

The number format can differ in features such as the decimal separator based on the locale; for example, the US formatting is 1,234,567.89 while German formatting is 1.234.567,89. The session currency is determined by the following, in order of consideration:

- Single-currency mode setup using glide.i18n.single_currency and glide.i18n.single_currency.code.
- The default currency for the user's locale.

### Reference currency

In order to perform calculations on heterogeneous currency values, the platform stores currency values converted to a system currency, referred to as the reference currency. Every currency field in the system contains a reference currency value. The reference currency is determined by the following, in order of consideration:

- The system locale set using the property glide.system.locale
• The Java default locale, typically en.US

The filtering and aggregation features use the reference currency value to perform calculations on default currency fields. This can yield inaccurate results because of conversion rate changes.

**Issues with currency fields**

Users are often confused by the results of filtering, sorting, and displaying currency fields because the system works with at least two currencies for each value: the session currency and the reference currency.

**Note:** Aggregations and filtering of currency fields use the reference currency, and the user sees the session currency. Because of changing conversion rates, the filtered reference currency values might not result in the same order as the session currency values would suggest. The same issue happens with aggregations.

The user might see the following issues:

• Lists filtered on currency fields might not be in the expected order because the reference currency values are used for filtering but session currency values are displayed.
• Aggregation of currency fields might not produce the expected results because reference currency values are aggregated and then converted to the session currency.
• Currency values might not be formatted as expected because currency values are formatted based on the user’s locale and not on the currency code

The confusion is caused by the difference between session and reference currencies, changing conversion rates, and different session currencies used by different users.

**Single-currency mode**

Single-currency mode allows all users of the platform to see currency values in the same currency.

Before enabling single-currency mode, the system locale must be set. See [Locale setting](#) for valid locale formats.

To configure single-currency mode, set the following properties:

• i18n.single_currency – true or false
• i18n.single_currency.code – the three-letter ISO currency code
• The system locale – glide.system.locale

Single-currency mode has the following limitations:

• Single currency mode changes the currency in the user views and does not change the number formatting. Even through users in different countries see currency values in one currency, the number formatting (as determined by the user’s locale) might not be what they expect.
• Because currency value input is constrained to the single currency, price fields cannot be used.

The effects of rate conversions can be avoided by setting the system locale and the reference currency to be the single currency.

**Price fields**

A price field is a currency field that enables control over conversions and display. The Service Catalog uses price fields.
The conversion and display selections can be chosen per price field and can be changed at any time. There are three variations:

- Calculated [Default]: Behaves the same as the currency field type. Whenever conversions are performed, the latest currency conversion rates are used. When the price field is displayed, it is shown in the user’s session currency.
- Fixed: When the price field is displayed, it is shown in the currency code used when the value was entered. Whenever conversions are performed, the latest currency conversion rates are used.
- Multiple: Enables you to enter multiple price values for an item using a different currency for each price. The field’s value is the value entered in the user’s session currency; otherwise, the first price entered is converted to the user’s session currency. Whenever conversions are performed, the latest currency rates are used. Note: The first value entered is used during display. The additional values are not used during calculations.

For examples of using price field, refer to the tables used in Service Catalog.

A price field’s currency code and numeric value can be changed in a form. An edit icon is shown next to the price field. Clicking the edit icon displays a form that can be used to edit all details of the price field:

- Currency: List of currencies enabled in the system in the combo box. In single currency mode, the currency is a label and cannot be changed.
- Amount: Numeric value formatted in the user’s locale
- Type: Combo box with calculated, fixed, multiple
  - When the price type is changed to multiple, the system creates child records for all currencies enabled in the platform populated with values converted from the amount field using latest currency conversion rates.
  - In single currency mode, the type cannot be changed.
  - The price type can be modified any time.

Currency values in lists

In lists, currency values are displayed in the user’s session currency formatted for display in the user’s locale. Typically the currency symbol is followed by a formatted number.

Different field types appear as follows:

- Currency field type: Value in user’s session currency
- Price field type/Calculated: Value in user’s session currency
- Price field type/Fixed: Value in currency as entered by the user
- Price field type/Multiple: Value associated with the user’s session currency if this value exists; otherwise, the first value entered is converted to the user’s session currency

A Globe icon is displayed beside the currency value that enables the value to be changed to one of the following values:

- Value as entered by the user
- Value in session currency
- Value as entered and, in brackets, the value in reference currency.

The icon appears when the user’s session currency is different from the currency entered. Clicking the icon cycles through the listed displays.

In the preview for the record, currency values are shown as entered, formatted for display in user’s locale.
**Aggregation**

Aggregation operations can be used on currency columns. Aggregation operations include total, group by, average, minimum, and maximum. Aggregation is done in two steps:

- Aggregate the reference currency values for all records
- Convert this aggregate to the user’s session currency for display

**Note:** Because the conversion rate between the currency field’s value (what is displayed) and its reference currency value (used for the aggregation) might have changed, the result may not be what the user expects.

This limitation extends to different price types.

- For price type fixed, the calculated reference value can be old.
- For price type multiple, the reference value of the first price entered is used. The other values are not used.

The aggregate value is shown formatted in user’s locale with a currency symbol.

**Filtering**

You can set up filters on currency fields. The currency value is entered as a currency code and numeric value. Filtering is done in two steps:

- The filter currency value is converted to the reference currency.
- The filter’s calculated reference value is compared with the reference value in the records.

Matching records are shown in the list view.

**Note:** Because the conversion rate used when the filter is run might be different than the conversion rate used when calculating the reference values in the individual records, filtering results might not provide the expected result.

This limitation extends to different price types.

- For price type fixed, the calculated reference value can be old.
- For price type multiple, the reference value of the first price entered is used. The other values are not used.

**Currency values in forms**

In forms, currency values are shown in the currency in which they were entered.

In forms, a combo box gives the list of currencies available in the system. The format is determined by the user’s locale. When entering or changing the numeric value, format the value in the format specified by the user’s locale. In the form for a new record, the combo box with the list of currencies has the reference currency selected, and the numeric value set to zero.

If the record is read only, the currency value is shown as entered and formatted for display in the user’s locale. A price field shows the session currency value.

In single currency mode, the currency is a label and cannot be changed. The form for editing the details of fields previously mentioned cannot be accessed because the edit icon is not shown.
Editing the currency instance table

For users who can edit the currency instance table (fx_currency_instance), an edit icon appears next to the field. Users with the financial_mgmt_user role can edit the values associated with the currency field.

**Note:** Do not edit the fx_currency_instance table directly. The platform maintains this table, and your changes could have unintended consequences.

Currency values in reports

Currency values in reports are in the user’s session currency formatted in the user’s locale with a currency symbol.

The user depends on how the report is run.

- Shared report: The user who runs the report
- Scheduled report: Generally run as the user who scheduled the report

The two user-specific values in the report are:

- User session currency
- Converted value

**Note:** A user that has a different session currency than the person who runs a report might receive unexpected results.

Currency conversions

Currency values may be converted to other currencies when stored and accessed.

Conversions may happen in these situations.

- The currency value is converted to reference currency when stored, whether on insert or update. This means that the reference currency value is saved as well as the currency value.
- The currency value is converted to the user’s session currency for display.
- The value entered for a filter from currency specified in the filter is converted to the reference currency.

Conversion rates are stored in the fx_rate table. Each record contains the conversion rate from a given currency to the Euro. The rates are updated daily from the ECB website by a scheduled job called ECB Exchange Rate Load.

A currency conversion from one currency to another involves two rates

- Rate to convert from the first currency to Euro
- Rate to convert from Euro to the second currency

Whenever a conversion is performed, the platform uses the latest conversion rates. Therefore, calculations can potentially yield unexpected results. For example:

- Different currency values can have different rates applied to them while storing the reference currency value. Aggregation therefore can combine values at different rates and convert back at another rate.
- A filter value is converted at current rates while the values it filters in the database can be converted at different rates. A filter for $100 at today's rate can match a value of $99 obtained at yesterday’s rates.
Set a conversion rate

ServiceNow downloads a currency conversion table from the European Central Bank nightly by default. You can adjust the frequency of this behavior or disable it entirely.

1. Navigate to System Scheduler > Scheduled Jobs.
2. Open the job named ECB Exchange Rate Load.
3. Modify the schedule, as needed.
   After the job runs, rates are stored in and loaded from the Exchange Rate [fx_rate] table. Navigate to System Localization > Exchange Rates to see them.

Use your own conversion table

ServiceNow bases all currency conversions on the rates stored in the Exchange Rate table.

1. Navigate to System Scheduler > Scheduled Jobs.
2. Open the job named ECB Exchange Rate Load.
3. In the Trigger type field, select -- None --.
4. Enter new exchange rates either manually or with an import set.

Currency values in import and export

In general, currency values crossing the boundaries of the platform are represented in the user’s session currency and formatted in the user’s locale.

Import

Currency values are imported as strings just like other fields. The default transform mapping to a currency field uses `setDisplayValue()`. The expected format for this function is:

- A number formatted in the user’s locale: this is taken as a value in the user’s session currency, for example, 1,234.56.
- The number prefixed by the three-letter currency code separated by a semicolon, for example, EUR;1,234,56.

This behavior can be customized in transform map scripts.

Export

Currency values are exported in the user’s session currency formatted in the user’s locale except when exporting as XML. When exporting currency in XML, the value is in the reference currency value with no formatting.

Currency values in scripts

You can use currency fields in scripts.
These methods are available on GlideElement objects.

To display currency values, use the display APIs (getDisplayValue()). To work with currency values in any way other than display, use the APIs that return/accept unformatted numbers.

**Note:** Do not use the getDisplayValue() methods and then process the string to remove formatting information before performing calculations on the value.

Methods such as getValue() and getCurrencyValue() return unformatted numbers as strings. The floating point value can be obtained by using the JavaScript function parseFloat(). The resulting value can be used to perform calculations. The currency associated with these values can be obtained by the APIs that return the currency code. You can also use the getCurrencyCode() methods to determine a field's currency.

```javascript
var rate = parseFloat(current.base_rate);
var currencyCode = current.base_rate.getCurrencyCode();
```

Use the setValue() method to set the value of a currency field. If the currency is the user's session currency, use a plain number (either floating point number of a string containing it), otherwise prefix the value with the 3-letter ISO currency code.

```javascript
var totalCost = rate*current.hourly_rate;
currency.total_cost.setValue(currencyCode + "\;\;" + totalCost);
```

When you use GlideAggregate on currency or price fields, you are working with the reference currency value. Be sure to convert the aggregate values to the user's session currency for display. Because the conversion rate between the currency or price field's value (what is displayed) and its reference currency value (used for the aggregation) might have changed, the result may not be what the user expects.

When a record containing a currency value is deleted, the platform deletes any associated currency records.

**Note:** Do not use deleteMultiple() on tables with currency fields. Always delete each record individually.

In the following table, the example values use a currency value of 21345.67 in Japanese yen (1563.72 in Euros and 1152.48 in US dollars) with the user's locale set to German (de.DE) and reference currency set to USD.

<table>
<thead>
<tr>
<th>Method name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>getValue()</td>
<td>Returns the currency value in the user's session currency as an unformatted number.</td>
<td>1563.72</td>
</tr>
<tr>
<td>getReferenceValue()</td>
<td>Returns the currency value in the reference currency as an unformatted number.</td>
<td>1152.48</td>
</tr>
<tr>
<td>getSessionValue()</td>
<td>Returns the currency value in the user's session currency as and unformatted number.</td>
<td>1563.72</td>
</tr>
<tr>
<td>Method name</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>getCurrencyValue()</td>
<td>Returns the currency value as entered as an unformatted number.</td>
<td>21345.67</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This might not be the session or reference currency.</td>
<td></td>
</tr>
<tr>
<td>getDisplayValue()</td>
<td>Returns the currency value in the user's session currency formatted in the user's locale with a currency symbol.</td>
<td>€1.563,72</td>
</tr>
<tr>
<td>getSessionDisplayValue()</td>
<td>Returns the currency value in the user's session currency formatted in the user's locale with a currency symbol.</td>
<td>€1.563,72</td>
</tr>
<tr>
<td>getReferenceDisplayValue()</td>
<td>Returns the currency value in the reference currency formatted in the user's locale with a currency symbol.</td>
<td>$1,152.48</td>
</tr>
<tr>
<td>getCurrencyDisplayValue()</td>
<td>Returns the currency value as entered formatted in the user's locale with a currency symbol.</td>
<td>¥21.345,67</td>
</tr>
<tr>
<td>getCurrencyString()</td>
<td>Returns the currency value as entered as an unformatted number prefixed by the 3-letter ISO currency code separated by a semicolon.</td>
<td>JPY;21345.67</td>
</tr>
<tr>
<td>getCurrencyCode()</td>
<td>Returns the 3-letter ISO currency code for the currency value as entered.</td>
<td>JPY</td>
</tr>
<tr>
<td>getSessionCurrencyCode()</td>
<td>Returns the 3-letter ISO currency code for the user's session currency.</td>
<td>EUR</td>
</tr>
<tr>
<td>getReferenceCurrencyCode()</td>
<td>Returns the 3-letter ISO currency code for the reference currency.</td>
<td>USD</td>
</tr>
<tr>
<td>setValue()</td>
<td>Sets the currency value as:</td>
<td>4369.21 or JPY;4369.21</td>
</tr>
<tr>
<td></td>
<td>• An unformatted number taken as a value in the user's session currency.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• An unformatted number prefixed by a 3-letter currency code separated by a semicolon.</td>
<td></td>
</tr>
</tbody>
</table>
### Currency properties

You can control how currency fields are configured and used in your instance. These properties are available for currency fields.

**Note:** To open the System Properties [sys_properties] table, enter `sys_properties.list` in the navigation filter.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>glide.system.locale</code></td>
<td>The value is of the format Language.Country where the language is an ISO 639 language code and the country is an ISO 3166 language code. Internally, this value is used as specified by Java. The system locale setting should be in the Java supported locales list.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> String</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> 4</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location:</strong> System Properties &gt; System Localization</td>
</tr>
<tr>
<td></td>
<td>• <strong>Learn more:</strong> <a href="#">Locale settings</a> on page 955</td>
</tr>
</tbody>
</table>
By default, the ServiceNow ITSA Suite uses US standard formatting (e.g., our current default is the US Dollar sign $ displayed with two decimal places: $100.00). By customizing your locale, you can make things such as currency appear as you expect. For example, in France, one might wish to see 100,00 € instead of $100.00.

Set the instance locale

Role required: admin

1. Navigate to System Properties > System Localization.
2. Enter the locale code to use under Locale code to use for localization. Format is [language code]. [country code] (e.g. en.GB for Britain fr.FR for France, de.DE for Germany, or ja.JP for Japan).

<table>
<thead>
<tr>
<th>Country</th>
<th>Locale code</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>en.US</td>
</tr>
<tr>
<td>Great Britain</td>
<td>en.GB</td>
</tr>
<tr>
<td>France</td>
<td>fr.FR</td>
</tr>
<tr>
<td>Germany</td>
<td>de.DE</td>
</tr>
<tr>
<td>Japan</td>
<td>ja.JP</td>
</tr>
</tbody>
</table>

3. Click Save.

Internationalization support

ServiceNow supports multiple languages, using UTF-8 for international characters.

When a user logs in, the system uses either the system default language or the language specified in the user record. This setting determines the language in which forms and messages are displayed throughout the session.

Language internationalization support

The ServiceNow system supports multiple languages, using UTF-8 for international characters.

When a user logs in, the language for the instance session is determined by the following logic:

1. If the language selection at login is enabled, that language is used.
2. If not, the language preference selected using the language picker in the header bar is used.
3. If not, the user’s language setting in the User [sys_user] table is used.
4. If none of the above are true, the system default language is used.

Some areas of the system are not translated, including journal fields, report titles, and any field that stores free-form text. These fields remain in the language used to create them.

When creating a custom field, you must add the labels in the Field Label table because they are not added automatically.
Table 243: FAQ

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Will my custom fields be translated?</td>
<td>When you create a field, it does not create a translated label. Labels for custom fields are created with a language of English (en). You can, however, translate the fields manually.</td>
</tr>
<tr>
<td>I am not seeing journal fields translated.</td>
<td>User-defined string fields, such as short description, additional comments, and report titles are not translated. They appear in the language they were created in.</td>
</tr>
</tbody>
</table>

Global language
You can set the default language.

Before selecting a new default language, activate the plugin for the desired language in System Definition > Plugins. The default global language for the system is set in System Properties > System Localization. This property defines the language that users see if a language is not specified in their user record.

![Default language for the system (two character values):](image)

User specific language
The user has several choices for identifying the language to be used for the instance, as described here.

- Language picker at login: If user-specific language is enabled, users see a choice list on the login page to select their language. To control display of the choice list on the login page, navigate to System Properties > UI Properties and use the property Show the language select box on the login page to allow the user to specify the language they would like to be logged in with.
- Language picker in the welcome banner: When an internationalization plugin is enabled, users can select their language in the language picker in the welcome banner.
- Language setting on the user table: If you have users that require a different language in addition to the global language, you can specify a language for them in User Administration > Users. If the Language field isn’t already on the form, you can configure the form and add it.

The Language field in the User [sys_user] table overrides the default global language for that user’s sessions.
Figure 309: User language

**Note:** Setting the language for the system guest user sets the language for both the login page and all users without a user role.

**Translation tables**
ServiceNow stores translation information in these tables.

- Languages [sys_language]
- Translated Name / Field [sys_translated]
- Message [sys_ui_message]
- Field label [sys_documentation]
- Choice [sys_choice]
- Translated Text [sys_translated_text]

**Note:** The Languages table is available only after I18N:Internationalization has been activated.

**Languages table**
The Languages [sys_language] table contains a list of the languages for which translated text is available.

To enable translation to a new language, add a record to the Languages table. The main fields for this table are:

- Name: language name.
- ID: two-character ISO 639.2 code for the language.
- Text Direction: direction of text in this language.
- Active: indicator that shows whether the language has been activated (true) for this instance or not (false).
Configure a language as reading from right to left
Use the Text Direction field to configure a language that reads from right to left, such as Hebrew.

Role required: admin

Right-to-left language support is available only in the main user interface and on live feed. Other user interfaces and applications, such as the graphical workflow editor, reporting, CMS, chat, and the ServiceNow documentation sites, are not supported.

1. Navigate to System Localization > Languages.
2. Click New.
3. Enter the Name of the language, such as Hebrew.
4. Enter the two-character ISO 639.2 ID for the language. For example, Hebrew is he.
5. In the Text Direction field, select Right-to-Left.
6. Click Submit.

Translated Name / Field table
The Translated Name / Field [sys_translated] table stores translated values for text fields where the field type is translated_field (see the dictionary entry).
This option is available for text fields up to 255 characters in length. Some examples are names, titles, and short descriptions. The main fields for this table are:

- Table: name of the table this translation applies to.
- Element: name of the field this translation applies to.
- Label (translate): translated text that users see on forms and lists.
- Language: two-character ISO language code for this translated text.
- Value: English value that causes this translated text to be displayed. For example, the first item in the illustration represents the text of a multiple choice answer in a survey. If the English text is D – Poor and the current ServiceNow session uses French, that choice appears as D – Faible.
The Message [sys_ui_message] table contains the translations for informational messages, confirmation messages, error messages, and other types of system messages.

ServiceNow checks this table for translated text when a client script contains a `getMessage` call or a server script contains a `gs.getMessage` call. The main fields for this table are:

- **Application**: name of the application this message appears in.
- **Key**: internal unique identifier of this message.
- **Language**: language the message is translated into.
- **Message**: translated text that users see.

**Figure 313: Message list**
Field Label table

The Field Label [sys_documentation] table stores the text of table names along with the singular and plural labels for each field in the table.

For each table name and field label, the Field Label table contains a record for each installed language. ServiceNow uses the table and field names from this table to display lists and forms in the proper language. The main fields for this table are:

- Table: name of the table this translation applies to.
- Element: name of the field this translation applies to.
- Language: two-character ISO language code for the translated text.
- Label: translated text that users see.
- Plural: plural of the label.
- Help: reserved for future use.
- Hint: text that pops up when the cursor rests on the field.
- URL: URL for a web page that provides information about the field. When a URL is provided, the field label displays a help icon ( ).
- URL target: location where the URL appears, if a URL is given. If this field is empty, the URL opens in the current tab or window when a user clicks the help icon. If the field contains the code _blank, the URL opens in a new tab or window when a user clicks the help icon.

The main fields for this table are:

- Table: name of the table this translation applies to.
- Element: name of the field this translation is used for.
• Language: two-character ISO language code for the translated choice.
• Value: English description of this choice.
• Label: translated text that users see for this choice.

Figure 317: Choice list

Figure 318: Translated choice list

Translated text

The Translated Text [sys_translated_text] table stores translations for fields with the field type translated_text or translated_html (see the dictionary entry).

This field type is typically used for long text fields, up to 6500 characters in length, such as survey name and introduction. The main fields for this table are:

• Document: internal identifier of the record this translation applies to.
• Field name: field this translated text appears in, for example, Close notes.
• Language: language the text is translated into.
• Table Name: table this translation applies to.
• Value: translated text that the user sees.
Show translated strings

When setting up translations for the different areas of your instance, a property can be turned on that allows you to easily determine which table you need to add the translated label to.

Role required: admin

1. Navigate to **System Properties > System Localization**.
2. Select Display translation prefix on translatable strings.

Displays translation prefix on translatable strings.

- Yes
- No

3. After activating the property, refresh your browser.
4. Following the refresh, the following prefixes appear on fields, labels, and messages that have been translated internally.
Translate new customizations
When using one of the Internationalization plugins, most of the fields in the instance are automatically translated. However, customizations are not translated automatically, and need to be translated by hand. In this case, it is best to locate the individual untranslated strings, and insert those translations manually.

Below are three tools for locating untranslated strings:

- Displaying prefixes for translatable strings
- Exporting untranslated strings
- Using the translate and learn property

Locate translatable strings

These tables contain translatable strings, described more at length in Internationalization support on page 965:

- Translated Name / Field [sys_translated]
- Message [sys_ui_message]
- Field Label [sys_documentation]
- Choice [sys_choice]

Display a translation prefix
Translation prefixes indicate where to find the string for translation.

Role required: admin

1. To enable prefixes on field labels for the current user session, navigate to System Localization > Enable I18N Debugging.
2. To disable prefixes for the current session, navigate to System Localization > Disable I18N Debugging.
• For the prefix TRF, navigate to **System Localization > Translated Names / Fields.**
• For the prefix MSG, navigate to **System Localization > Messages.**
• For the prefix GMLD, navigate to **System Localization > Field Labels.**
• For the prefix TRT, navigate to **System Localization > Translated Text.**
• For the prefix CHC, navigate to **System Localization > Choices.**

Some few strings may not display translatable prefixes. This means that the string is not stored on any of these four tables. This behavior occurs with text embedded in images, such as the buttons in the Service Catalog, or text defined by properties, such as the text which follows the banner.

You must refresh your browser after accessing one of these modules to apply the change.

**Export an untranslated string**

One method for easily translating customizations is to export all of the translated names and fields, messages, field labels, and choices that only have English translations.

Role required: admin

To aid in this, there are four modules for the **System Localization** application menu that are inactive by default.

• Customizations
• Translated Name / Fields
• Messages
• Field Labels
• Choices

1. Perform the appropriate action for your version of the UI:

| UI16 | 1. Navigate to **System Definition > Application Menus.**  
|      | 2. Select **System Localization.** |
| UI15 or UI11 | Right-click the **System Localization** application menu and select **Edit Application Menu.** |

2. Use the list editor to set the **Active** field to **True** for the modules.
After the application navigator refreshes, the modules appear. Now it is possible to export the list of untranslated fields by viewing each of the lists and exporting it to any supported format.

Use the translate and learn property

The glide.translate.learn system property, when set to true, creates records in the translate tables when an instance encounters text that should be translated, but no corresponding record exists. You can use this property to help locate untranslated strings.

**Note:** Using this method hampers the use of the exporting untranslated strings method.

For example: A user switches to the French language and opens a record. When loading the form, the instance looks at the Translated Field table to find the French translation for each field. If an appropriate translation record is not found, the instance automatically generates a stub record for that missing data, which can be manually translated.

The following HR application is missing translations for two modules:

![Image](image.png)

**Figure 320: HR application missing translations**

The suffix [fr] indicates that there is a French translation missing. Because translation prefixes are enabled, the prefix **TRF:** indicates that the entry can be found in the Translated Name / Fields table. The following image shows untranslated modules, located using the filter **[Label (translate)] [contains] [fr]:**

![Image](image.png)

**Figure 321: Example untranslated modules**

**Service catalog buttons**

You can specify language-specific messages for buttons in these service catalog screens: Cart, Edit Cart, and Check Out (including workflows and approvals).

The text for the buttons is stored on the "Messages" table.
Activate a language

By default, the language supported by the platform is American English. You can activate other supported languages if you have the admin role.

The following plugins are currently available:

- I18N: Brazilian Portuguese Translations
- I18N: Czech Translations
- I18N: Dutch Translations
- I18N: Finnish Translations
- I18N: French Canada Translations
- I18N: French Translations
- I18N: German Translations
- I18N: Hebrew Translations
- I18N: Hungarian Translations
- I18N: Italian Translations
- I18N: Japanese Translations
- I18N: Korean Translations
- I18N: Polish Translations
- I18N: Portuguese Translations
- I18N: Russian Translations
- I18N: Simplified Chinese Translations
- I18N: Spanish Translations
- I18N: Thai Translations
- I18N: Traditional Chinese Translations
- I18N: Turkish Translations
In addition, the I18N: Internationalization plugin provides the elements necessary for translating an instance without any translation preloaded. This plugin is useful for translating an instance to a language other than those listed above. For more information on using the I18N: Internationalization plugin to translate an instance, see Translate an instance.

**Note:** Activating internationalization plugins for any of the available languages automatically activates the I18N: Knowledge Management Internationalization Plugin v2 plugin (com.glideapp.knowledge.i18n2).

Use the following steps to activate the desired language plugins.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

**Localize price in the service catalog**

The ServiceNow service catalog fully supports localized currencies for item prices and options.

**Note:** ServiceNow has the concept of a system basis currency: the system default currency, based on the global locale property. All currency values are automatically converted to this basis currency before aggregation or conversion. Do not change the global locale setting after you have data in the system, as this can cause aggregations to calculate and display incorrectly under some circumstances.

ServiceNow supports two similar but different money field types, both of which are used in the service catalog.

**Price:**

A price represents the cost of the catalog item. A computer might cost $1000, or the provisioning of an email account may require an $80 chargeback.

ServiceNow supports these pricing models:

- **Calculated:** [Default] the price of an item is always quoted based on the user's locale. For example, a UK user sees the price of her computer in pounds, while a Japanese user sees his price in yen. The numeric value of the price is adjusted using the most current exchange rates on file.

  The locale is based on the Country code field [sys_users.country] on the user record. If this field is not set, the locale is taken by default from the instance locale, as defined by the locale code to use for localization [glide.system.locale] property.

- **Fixed:** the price of an item is always quoted in a particular currency. An item priced at $1,000 is always priced at $1,000, even if viewed by a UK or Japanese user.

- **Multiple:** a fixed price is specified for each currency. For example, you can specify that all US users pay USD 1000 for a computer, while the Japanese price is JPY 120,000, regardless of exchange rate.

**Note:** If no fixed price is set, ServiceNow uses the calculated price (default).
Currency:

A currency field represents an actual spend, that is, a discrete quantity of money spent at a particular point in time. ServiceNow keeps track of the currency that was actually spent, and the time it was spent.

On a related list, a currency field uses one of these modes. Toggle between them by clicking the globe icon next to the currency field.

- **As spent**: displays the amount in the currency that was spent. This may mean that some items in the list show yen, while others show pounds or dollars, for example, £100.00.
- **Calculated**: displays all amounts in the session display currency, for example, $137.86.
- **Reference**: displays all amounts in the currency entered and includes the system basis currency in square brackets, for example, £100.00 [$137.86].

On a form, a currency field always displays the amount in the currency entered.

**Note**: If you reference a currency field in script, its reported value is whatever the current user’s session would show. There are also several utility functions you can call against a currency field to get data back in a variety of forms. For more information, see *Scripting Currency and Price Fields*.

### Calculated currency

For price fields using the calculated currency model, the system converts the price you enter into whatever is appropriate for a particular user’s locale.

For example, in this case we’re specifying a calculated currency and we’ve entered the value of $500.

![Figure 323: Bb calculated](image)

**Figure 323: Bb calculated**

If we then view our catalog item as a UK user, we’ll see a price in pounds, in this case £362.67.

![Figure 324: Bb calculated pounds](image)
Fixed currency

For price fields using the fixed currency model, the price displayed is in the specific currency that you select.

In this example, we’ll specify that the price of the Blackberry should always be quoted in Euros.

![Currency Price](image)

**Figure 325: Currency Price**

If we view the item as a UK user, we see Euros. Note that users’ carts always display in their session currency, so the blackberry that was placed in this cart is priced in GBP.

![Catalog Item Price](image)

**Figure 326: Catalog Item Price**

Multiple currency model

For price fields using the multiple currency model, you specify a fixed price based on locale.
For example, in this model, we specify that the item costs USD 500, but GBP 450, regardless of current exchange rates.

Figure 327: Multiple currency price

If we then proceed to view this as a UK user, instead of seeing a price of about 365 pounds (which is what we’d get if we converted USD 500 into pounds at the time of this writing), we instead see the price of exactly GBP 450.

Figure 328: Catalog item fixed price

Display currencies to catalog users

You can specify that an item and its options use any of the pricing models described.
For example, set a fixed price for the default iPhone item in pounds (£), and it always appears in pounds (£), even for a US-based user. The same is true for variables with pricing implications. If you specify a fixed price on the options, then ServiceNow displays the option in that currency on the form for ordering the item.

However, prices of items displayed to end users in the shopping cart and checkout screens are always shown in the logged-in user’s session currency. For example, if a US-based end user adds a £100.00 item to his cart, the shopping cart shows the equivalent value in US dollars.

**Reporting**

Currency and price fields can be queried and reported.

Reports and aggregations always display values in the reference currency.

**Default Reports**

ServiceNow provides two reports that address service catalog spending:

- **This Year’s Spending by Department**: spending for the previous 12 months, broken down by month and department.
- **This Year's Spending by Location**: spending for the previous 12 months, broken down by month and requester location.

To run these reports, navigate to **Reports > View / Run** and go to the Requested Item section.

**Creating Custom Reports**

When creating custom reports, consider:

- Reports display currency values converted to the currency of the user running the report. A shared report displays in Euros when run by a German, but in US dollars when run by an American. Currency conversions do apply.
- Scheduled reports generally run as the user who scheduled them. So a report scheduled by an American and emailed to three Europeans shows values in US dollars.

For more information, see **Create or edit a report**

**Use locales**

ServiceNow uses the language and country specified in the user record to determine the currency and pricing model for a particular user.

If these records are incomplete or incorrect, currency and pricing models are based on the default locale. For more information, see **Define locales** on page 964.

**Set prices**

The Price field on catalog items uses the price data type, meaning that catalog items and lists of catalog items show localized prices.

By default, price fields use the calculated pricing model, meaning that regardless of the currency used to enter the price, ServiceNow converts that value to the user’s session currency and displays that converted value to the user.

When defining catalog items, catalog administrators can specify a currency and a value for the item, and can alter the pricing model by clicking the **Edit** link beside the Price field on the Catalog Item form.
Set a conversion rate

ServiceNow downloads a currency conversion table from the European Central Bank nightly by default. You can adjust the frequency of this behavior or disable it entirely.

1. Navigate to System Scheduler > Scheduled Jobs.
2. Open the job named ECB Exchange Rate Load.
3. Modify the schedule, as needed.

After the job runs, rates are stored in and loaded from the Exchange Rate [fx_rate] table. Navigate to System Localization > Exchange Rates to see them.

Use your own conversion table

ServiceNow bases all currency conversions on the rates stored in the Exchange Rate table.

1. Navigate to System Scheduler > Scheduled Jobs.
2. Open the job named ECB Exchange Rate Load.
3. In the Trigger type field, select -- None --.
4. Enter new exchange rates either manually or with an import set.

Use translated text

Each instance can be localized, translating the instance to the instance's local language.

Translated text fields allow the same field to display different content based on the user's language.

There are two different translated text fields:
- Translated Text
- Translated HTML

These fields operate the same as text and HTML fields respectively, except that they can store multiple inputs in multiple languages.

The most frequent uses of translated text fields are in the Knowledge Base (e.g. article titles or content) or the Service Catalog (e.g. names, descriptions, or variables). The Knowledge Base also has an option for internationalization with the Knowledge Management Internationalization Plugin, which allows for language-specific articles rather than translating articles.

Note: The type is translated separately from Translated Text and Translated HTML fields.

Administer translated text fields

There is a slight performance penalty associated with changing a normal HTML or text field into a translated HTML or text field. It is best only to use translated fields if the translated capability is required.

English language text is stored in the master table, but the values of other are stored in the sys_translated_text table. Each translated field on every row has one or more entries in the sys_translated_text, one per language for which ServiceNow provides a translation.

Key Fields in the Translated Text [sys_translated_text] Table:
- tablename -- the table to which this translation belongs, e.g., problem
- fieldname -- the field to which this translation belongs, e.g., workaround
- documentkey -- the sys_id of the row to which this translation belongs, e.g., the sys_id of PRB00008
Translate the content of a translated text field

After a translated text field or translated HTML field has been created and is in use, it displays English when viewed in a different language until the content is translated.

Be sure to confirm that the field is in fact a translated text field, and not a regular text or HTML field, for example, by right-clicking the field and choosing Personalize Dictionary.

The method below works best for one-off translations. To translate large numbers of translated text fields, use the Exporting Untranslated Strings method.

1. Use the language picker to switch to the language the field's content is being translated to.
2. Navigate to the field on the form.
3. Replace the English text with the text of the target language.
4. Submit.

Now the text will display English when the user's language is English and in the new language when the user's language is set to that new language.

System localization

Localization allows administrators to accommodate users from a variety of different countries, using different languages and currencies, within the same instance.

The platform supports internationalization of language and localization of currencies and prices wherever they appear.

Set localization properties

System properties to localize the instance for users from multiple countries.

Role required: admin

Some localization properties only appear after the I18N: Internationalization plugin is activated.

1. Navigate to System Properties > System Localization.
2. Set these properties

<table>
<thead>
<tr>
<th>Table 244: Customization properties for system localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show the language select box on the login page to allow the user to specify the language they would like to be logged in with.</td>
</tr>
<tr>
<td>glide.ui.login.language.select</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Feature</td>
</tr>
<tr>
<td>---------</td>
</tr>
</tbody>
</table>
| **Text Search stemming language.** <br>glide.ts.stemming_language | Selects the language to match derived words in text search.  
- **Options**: English and German  
- **Default value**: English  
- **Dependency**: The I18N: Internationalization plugin must be active  
- **Learn more**: Zing matches derived words with stemming on page 1107 |
| **Displays translation prefix on translatable strings.** <br>glide.ui.i18n_test | Adds a translation prefix on translatable fields to indicate where to find the string for translation.  
- **Options**: Yes | No  
- **Default value**: No  
- **Learn more**: Display a translation prefix on page 974 |
| **Use a single currency model. Display all currencies in the same currency code, regardless of a user’s locale, country, or language code.** <br>glide.i18n.single_currency | Enable single currency mode.  
- **Options**: Yes | No  
- **Default value**: No  
- **Learn more**: Currency administration on page 955 |
| **Default language for the system (two character values)** <br>glide.sys.language |  
- **Options**: en, fr, es, it, de  
- **Default value**: en |
| **If using the single currency model, display all currencies using this currency code. Currency codes use the ISO 4217 three letter format.** <br>glide.i18n.single_currency.code |  
- **Options**: ISO 4217 three-letter currency code  
- **Default value**: USD  
- **Learn more**: Currency administration on page 955 |
Locale code to use for localization.
glide.system.locale

Determines the system's default currency into which all prices are automatically converted before other sums or conversions are performed. Changing this property after any price or currency fields have been given a value may result in improper conversion or prices that sum incorrectly.

- **Options**: [language code].[country code] (for example, en.GB for Britain, fr.FR for France, de.DE for Germany, or ja.JP for Japan)
- **Default value**: None
- **Learn more**: Locale settings on page 955

---

Add the labels, messages, or choices to the appropriate table in English with an ending of the language code for newly added customizations that are missing translations. (Translate and Learn).
glide.translate.learn

Adds the current language suffix to UI elements such as labels and messages. Enabling the language suffix assists with translating new customizations.

- **Options**: Yes | No
- **Default value**: No
- **Dependency**: The I18N: Internationalization plugin must be active
- **Learn more**: Use the translate and learn property on page 976

---

Spell checker dictionary to use for English users.
glide.spell.dictionary.en

Sets the spell checker dictionary used in the system for English users.

- **Options**: Brazilian Portuguese, Dutch, English US, English UK, French, German, Italian, Portuguese, Russian, Spanish, and Thai
- **Default value**: English US
- **Learn more**: Locale settings on page 955

---

**Currency localization**

Currencies can be defined within the System Localization application, and can have their exchange rates loaded from the European Central Bank.

Once the Currency Support for Service Catalog Plugin has been installed, the service catalog can convert catalog item prices into the currency of the user's locale code, or can display a fixed price that will display in the same currency regardless of who is viewing it.

© 2017 ServiceNow. All rights reserved.
Location setup

If your organization supports more than one distinct location, you can configure these locations in the ServiceNow ITSA Suite to help further identify users, assets, incidents, and so on.

To see available locations for your company, navigate to **User Administration > Locations**. Click the New icon (![](images/new_icon.png)) to add a new location:

![Figure 329: Adding a new location](images/location_setup.png)

When done, click the Submit icon (![](images/submit_icon.png)). Your new location will now be available to you anywhere you can specify one.

Translate an instance

ServiceNow provides a series of Internationalization plugins, each of which translate most of the instance into a particular language.

Administrators can also translate an instance into languages other than those provided in the internationalization plugins.

**Activate the I18N: Internationalization plugin**

The first step in translating an instance is to install the elements required for translation, including tables to hold the translations, language pickers to allow users to switch between languages, and import set tables and transform maps to aid in importing translations. These elements are all provided in the plugin I18N: Internationalization.

**Request Survey Wizard**

**Create a new choice record**

This record allows users to select the language as a valid option in a User record and the language picker.
You must create a choice record for a new translation in the Choices [sys_choice] table.

1. Navigate to System Localization > Choices.
2. Click New.
3. Enter the following fields.
   • Table: Enter sys_user.
   • Element: Enter preferred_language.
   • Language: Enter the two-character ISO 639.2 code for the language this choice record is a member of. For example, tr. The default is en.
   • Label: Enter the name of the language selection as you want it to appear in the language picker. For example, Turkish.
   • Value: Enter the two-character ISO 639.2 code for the new language selection. For example, tr. The instance uses this value to set the display language.
   • Sequence: Enter a number to determine what order the option appears in the choice list if you do not want to list choices alphabetically. For example, 5.
4. Click Submit.

Create a new language record
You must create a language record for your new translation in the Languages [sys_language] table.

Role required: admin
1. Navigate to System Localization > Languages.
2. Click New.
3. Enter the following fields.
   • Name: Enter the name of the language. For example, Turkish.
   • ID: Enter the two-character ISO 639.2 code for the language. For example, tr.
   • Text Direction: Select the direction that the instance should display the language in. For example Left-to-Right.
4. Click Submit.

Import a translation from an Excel spreadsheet
The System Import Sets application contains four import tables and corresponding transform maps to assist with importing translations from an Excel spreadsheet.

1. Navigate to System Import Sets > Load Data.
2. Select Use Existing and the Table name that matches the type of data being imported, as follows.
   • For choices, select the [u_sys_choice] table.
   • For field labels, select the [u_sys_documentation] table.
   • For translated names and fields, select the [u_sys_translated] table.
   • For messages, select the [u_sys_ui_message] table.
   • For translated text, select the [u_sys_translated_text] table.
3. Select Upload an Excel file, and then click Browse to select the source Excel spreadsheet.
4. If appropriate, specify the Work sheet and Header row number.
5. Click Go.

The translations are now available in the appropriate Import Set Table.
6. Navigate to System Import Sets > Table Name and review the imported information to verify that the import was successful.
7. To transform the imported data into the corresponding table, navigate to **System Import Sets > Run Transform**.

8. Select the appropriate transform map, as follows.

    **Note:** Make sure you choose a transform map that has the **Run Business Rule** option selected. If the transform map does not have this option selected, any customized translations you have may be overwritten during the next upgrade.

    - For choices, select the **Sys Choice Translation Map**.
    - For field labels, select the **Sys Documentation Translation Map**.
    - For translated names and fields, select the **Sys Translated Translation Map**.
    - For messages, select the **Sys UI Message Translation Map**.
    - For translated text, select the **Sys Translated Text Translation Map**.

9. Click **Transform**.

*Translate a client script message*  
Client scripts include a multi-line Messages field.

Use this field to enter message strings that the client script can use as a key to look up a localized message alternative from the Message [sys_ui_message] table. Add each message key on a separate line. The instance looks for a localized message string anytime the client script makes a getMessage(msg) call where the msg string matches a key in the Messages field.

For example, if you add the string **Please populate the Reason field** to the Messages field, then the instance will look for a localized string from the Message [sys_ui_message] table any time the client script calls:

```
getMessage(Please populate the Reason field)
```

Add a new record to the Message [sys_ui_message] table for each localized string.

1. Navigate to **System Localization > Messages**.

2. Click **New**.

3. Enter the **Message** fields for the localized message.

4. Click **Submit**.

*Translate a field label*  
Field labels are the names that appear on forms and lists to describe the type of information the field contains.

The following procedure works best for translating individual field labels, such as those added with a customization. To translate large numbers of field labels, use the procedure described in **Translate the Interface**.

1. Navigate to the field on the form.

2. Right-click the field label and select **Configure Label**.

3. In the Field Label form, replace the English text with the text of the target language in the Label, Plural, and Hint fields.

4. Enter the two-character Language code of the target language.

5. Right-click the header bar and select **Insert**.

   Clicking **Insert** creates a new record in the Field Label table for this field label in the selected language.
Translate a field value
Field values are the text entries that are used for fields with the type translated_field, such as the Title or Hint field in the Module \([\text{sys\_app\_module}]\) table.

The following procedure works best for translating values for individual fields, such as those added with a customization. To translate large numbers of field values, use the procedure described in \textit{Translate the Interface}.

1. Use the language picker in the header bar to switch to the desired language.
2. Navigate to the field on the form.
3. Enter the text of the target language for this value.
4. Click \textit{Submit}.

   This creates a new record in the Translated Name / Field table for the selected language or updates the existing record.

Translate a related list name
Related lists appear at the bottom of forms.

You can translate a related list name by \textit{configuring the list}.

1. Use the language picker in the header bar to switch to the desired language.
2. Navigate to the related list on the form.
3. Right-click the related list header bar and select \textit{Configure > List Control}.
4. Replace the existing Label with the text of the target language.
5. Click **Submit** or **Update**.

This creates a new entry in the Translated Name / Field \[sys_translated\] table or updates the existing entry for this language.

*Translate long text content*

Long text content occurs in fields with the type translated_text or translated_html.

Use the following procedure to translate the content for individual text fields. To translate large numbers of text or HTML fields, use the procedure described under **Translate the Interface**.

To translate the content of a text or HTML field on a form (field type translated_text or translated_html):

1. Use the language picker to switch to the desired language.
2. Navigate to the field on the form.
3. Replace the English text with the text of the target language.
4. Click **Submit**.

This creates a new record in the Translated Text table for the active language. The field content displays English or the new language, depending on the user's language selection.

*Translate the interface*

After creating the choice record, translate the interface, including the applications, modules, UI actions, forms, lists, alerts, and choice lists.

All of these interface items are stored as translatable strings in these **translation tables**.

- Translated Name / Field \[sys_translated\]
- Messages \[sys_ui_message\]
- Field Label \[sys_documentation\]
• Choice [sys_choice]
• Translated Text [sys_translated_text]

By default, these translation tables only contain English strings. To populate these tables with translated strings:

1. Export the contents of the translation tables into a format (such as Excel) that can be easily manipulated.
2. Within the exported document, translate the Label, Plural, Hint, and Message columns for each row. Be sure to also change the "Language" column to the two character ISO code of the new language.
3. Import the translated document back into the instance as an import set.

Translate individual field labels and values
When translating just a few field labels or values, such as when you add customizations to a translated instance, use the procedure that applies to the type of text being translated.

Three types of ServiceNow fields store translated strings:

• Translated_field: Stores field labels, related list names, and certain field values. The value of the translated_field replaces the label, list name, or field value when the user selects the matching language. Translated_field values have a one-to-many relationship with their associated keys. As a result, multiple records can reference one translated_field value.

• Translated_text: Stores long text values in plain text. The value of the translated_text replaces the plain text when the user selects the matching language. Translated_text values have a one-to-one relationship with their associated keys. As a result, only one record can reference a translated_text value.

• Translated_html: Stores long text values in HTML. The value of the translated_html replaces the HTML when the user selects the matching language. Translated_html values have a one-to-one relationship with their associated keys. As a result, only one record can reference a translated_html value.

All three translated field types support list sorting. To determine the field type, right-click the field on the form, select Configure Dictionary, and check the Type field.

ServiceNow stores the translated values as separate records and displays the proper value according to the end user's language. You can translate an entire instance by exporting the translation tables and then importing the translated strings as described under Translate the Interface.

Note: In addition to translated field types, currency fields display the same price in different currencies based on the user's language.

Translate the knowledge base
The knowledge base has two separate methods for translation.

Translating the content of articles: used for articles that apply to users of all languages.

Creating language-specific articles: used when users with different languages need different articles.

Translating knowledge base articles

Knowledge base articles use translated_html fields for article content. This type of field displays the translation based on the user's language, if multiple translations are stored. To learn about using translated_html fields, see Use Translated Text.
Creating language-specific articles

Activating the *Knowledge Management Internationalization v2 plugin* adds a Language field on the Knowledge form for setting the language of the article. Users can choose a language to search and search results return only articles in that language. Users can switch between different translations of the same article, as well.

Time configuration

Schedule events and track time across applications.

Date and time fields

Records can store date and time values in several different types of fields.

These values are stored in the database as integer numbers of milliseconds, and are displayed in the appropriate date or time format.

Date and time field types

The following field types are provided for storing date and time information in records.

<table>
<thead>
<tr>
<th>Field type</th>
<th>Dictionary XML type</th>
<th>MySQL DB type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>glide_date</td>
<td>DATE</td>
</tr>
<tr>
<td>Date-Time</td>
<td>glide_date_time</td>
<td>DATETIME</td>
</tr>
<tr>
<td>Time</td>
<td>glide_time</td>
<td>DATETIME</td>
</tr>
<tr>
<td>Duration</td>
<td>glide_duration</td>
<td>DATETIME</td>
</tr>
<tr>
<td>Due-Date</td>
<td>due_date</td>
<td>DATETIME</td>
</tr>
</tbody>
</table>

Global date and time field format

The default date and time formats are defined globally using system properties.

**Date format**

The date format is defined by the property glide.sys.date_format.

An administrator can modify the property by navigating to *System Properties > System*. Modifying the property changes the date or time format globally. When modifying the standard date format, also verify the format using a *Validate Date and Time* script. Use the same 'format' strings as the java.text.SimpleDateFormat class, with minor exceptions. Note that MM is months, where mm indicates minutes. The format string consists of the following abbreviations.

<table>
<thead>
<tr>
<th>Field</th>
<th>Full form</th>
<th>Short form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>yyyy (4 digits)</td>
<td>yy (2 digits), y (2 or 4 digits)</td>
</tr>
<tr>
<td>Month</td>
<td>MMM (name or abbr.)</td>
<td>MM (2 digits), M (1 or 2 digits)</td>
</tr>
</tbody>
</table>
### Day of Month

<table>
<thead>
<tr>
<th>Field</th>
<th>Full form</th>
<th>Short form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day of Month</td>
<td>dd (2 digits)</td>
<td>d (1 or 2 digits)</td>
</tr>
</tbody>
</table>

The default format is: yyyy-MM-dd.

**Note:** A user can override the global date or time format with a personal preference.

*Personalize the system date format*

You can personalize the format in which date values appear in your instance.

Personalizing the date format does not change global settings or impact the way other users see date values.

1. Navigate to **Self-Service > My Profile**.
2. In the **Date format** field, select an option.
3. Click **Update**.

### Time format

The time format is defined by the property glide.sys.time_format.

An administrator can modify the property by navigating to **System Properties > System**. Modifying the property changes the date or time format globally. When modifying the standard time format, also verify the format using a **Validate Date and Time** script. Use the same 'format' strings as the java.text.SimpleDateFormat class, with minor exceptions. The format string consists of the following abbreviations.

<table>
<thead>
<tr>
<th>Field</th>
<th>Full form</th>
<th>Short form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hour (1-12)</td>
<td>hh (2 digits)</td>
<td>h (1 or 2 digits)</td>
</tr>
<tr>
<td>Hour (0-23)</td>
<td>HH (2 digits)</td>
<td>H (1 or 2 digits)</td>
</tr>
<tr>
<td>Minute</td>
<td>mm (2 digits)</td>
<td>m (1 or 2 digits)</td>
</tr>
<tr>
<td>Second</td>
<td>ss (2 digits)</td>
<td>s (1 or 2 digits)</td>
</tr>
</tbody>
</table>

The default format is: HH:mm:ss.

**Note:** A user can override the global date or time format with a personal preference.

*Personalize the system time format*

You can personalize the format in which time values appear in your instance.

An administrator must add the **Time format** field to the Self-Service view of the User form. For more information, see **Configure a form** on page 703.

Personalizing the time format does not change global settings or impact the way other users see time values.

1. Navigate to **Self-Service > My Profile**.
2. In the **Time format** field, select an option.
3. Click **Update**.
Configure the date picker for the list editor

In UI16 and UI15, a system property enables you to choose between two date picker configurations for the list editor.

Role required: admin

1. Navigate to `sys_properties.list`.
2. Search for the property named `glide.ui.list_edit.show_calendar_only`.
3. Set the property Value to either of the following options.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>The date picker displays a calendar as well as a field for manual date entry. This is the default behavior in UI11, regardless of the property value.</td>
</tr>
<tr>
<td>true</td>
<td>The date picker displays a calendar only. This is the default behavior in UI16 and UI15.</td>
</tr>
</tbody>
</table>
Default date and time fields

Certain time fields are provided by default to store particular date and time fields.

Global timestamp fields
All records inherit certain time stamp fields from the Global [global] table.
All records inherit the following time stamp fields from the Global [global] table.
- Created
- Updated

These fields are automatically populated with the correct date and time.

Task fields for measuring work time
Use default task fields to measure progress and resolution for certain records.

The following base system fields are provided on certain tables for keeping track of how long it takes to close tickets:
- **Time worked**: A timer which runs while the record is being viewed by a user, and pauses while the record is closed (or when it is paused manually). Used to keep track of the time spent by the help desk while working on the record.
- **Resolve time**: A calculated field which measures the time from the moment the record is opened, to the moment the record is closed. Used to keep track of how long it takes to resolve the record.

These fields provide different metrics for request response.

The following additional tools are available for tracking work time:
- **Service level agreements (SLAs)**: measure how long it takes a record fulfill certain conditions (such as an incident being marked **Resolved**).
- **Time cards**: use the **Time worked** field to break down how much time was spent by day of the week.
Time worked

The Task [task] table provides a time-tracking field called **Time worked**.

![Time worked: 03:13:18 / 00:00:10](image)

**Figure 330: Time worked field**

This field measures how long a record has been viewed in order to measure work time on a ticket. Any table that extends Task can use this field. To add the field, configure the form.

As the record is viewed, the timer counts upward. To pause the timer, click the stop icon (●).

To resume the timer, click the start icon (●).

When the task is saved, the amount of new time in the timer is used to generate a record on the Time Worked [task_time_worked] table. This table can be viewed as a related list on the task form.

By default, the time displayed in the **Time worked** field displays a cumulative value stored in the task record. If you modify a Time Worked record, the changes will not be reflected in the task timer.

You can set the property com.snc.time_worked.update_task_timer to enable updating of the task timer value based on changes to the time worked records. This is accomplished through the Update task timer business rule.

Resolve time

The **Resolve time** field is available on the Incident [incident] and Request [sc_request] tables.

Business rules calculate the field when the record is marked closed, and measure the difference between the **Opened** and **Closed** dates. This field allows for easy reporting on how long it takes for requests to be closed.

The field is stored in the system as an integer number of seconds.

**Business rule calculation:**

On the Incident table, the field is calculated on closure by the business rule mark_closed. The following lines of code calculate the resolve time:

```java
current.calendar_stc =
    gs.dateDiff(current.opened_at.getDisplayValue( ),
                current.closed_at.getDisplayValue( ), true);
```

On the Request table, the field is calculated on closure by the business rule Mark Request Closed. The following line of code calculates the resolve time:

```java
current.calendar_stc =
    gs.dateDiff(current.opened_at.getDisplayValue( ),
                current.closed_at.getDisplayValue( ), true);
```

Display resolve time as a duration

You can display the resolve time as a human-readable duration rather than an integer representing a number of seconds.

1. From the Incident or Request form, right-click the form header and click **Configure > Table**. The system displays the table form.
2. From the **Columns** embedded list, click **Resolve time**. The system displays the dictionary entry for the field.
3. From the **Attributes** related list, click **New**.
4. Enter these values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Format</td>
</tr>
<tr>
<td>Dictionary entry</td>
<td>Resolve time</td>
</tr>
<tr>
<td>Value</td>
<td>glide_duration</td>
</tr>
</tbody>
</table>

5. Click Submit.

Forms and lists display the resolve time as a number of days, hours, and minutes.

Note: This attribute does not change the field data format which remains an integer representing a number of seconds. Reports and data exports will still display the actual number of seconds rather than a duration.

Planned task time fields

The Planned Task plugin provides a table (Planned Task [planned task]) with standard fields for measuring a planned task's time. For more information, see Planned Task.

Export date and time formats

Because some export formats are intended for human consumption and others are intended for database usage, different methods provide date and time field information in different formats.

Excel

Date, Date-Time, and Time fields are all exported as their display values, displayed using a custom format instead of the system date format.

Duration fields, however, export as the value stored in the database, which is an integer value of seconds.

XML

All Date and Time fields export as the value stored in the database.

PDF

All Date and Time fields (including Duration) export as their display value.

CSV

All Date and Time fields export as the value stored in the database.

Calendars with schedule pages

Specific applications within the platform generate graphical calendar displays based on Schedule Pages.
These Schedule Pages can be displayed in a daily, weekly, or monthly view. Currently, the applications using Schedule Pages include:

- Project Management
- Maintenance Schedules
- Group On-Call Rotation
- Field Service Management

Schedule Pages are records which contain the scripts that determine the functionality of the graphical display. Because of the heavy degree of scripting involved in a schedule page, most instances should use the default schedule pages in the base platform. The schedule page uses a URL with a series of parameters attached to generate the graphical display.

Calendar content is controlled by a Schedule Page (cmn_schedule_page) record. To access Schedule Pages, navigate to System Scheduler > Schedules > Schedule Pages. The Schedule Pages form provides the following fields:

The Schedule Page record contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>General name that is used to identify the current schedule page.</td>
</tr>
<tr>
<td>Schedule type</td>
<td>String</td>
<td>The schedule type is a string that is used to uniquely identify the schedule page via the &quot;sysparm_page_schedule_type&quot; URI parameter. For example, a schedule page could be accessed as follows: /show_schedule_page.do?sysparm_page_schedule_type=gantt_chart&amp;sysparm_timeline_task_id=d530bf907f0000015ce594fd929cf6a4 Alternatively, the schedule page can also be accessed by setting the &quot;sysparm_page_sys_id&quot; URI parameter to that of the unique 32 character hexadecimal system identifier of the schedule page.</td>
</tr>
</tbody>
</table>
| View Type     | Choice     | Each view type displays different field combinations. There are two options available:  
- **Calendars**  
- **Timelines** |
<p>| Description   | String     | General description that provides additional information about the current schedule page. This field is not necessary. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init function name</td>
<td>String</td>
<td>The init function name specifies the name of the JavaScript function to call inside the Client script function for calendar type schedule pages.</td>
</tr>
<tr>
<td>HTML</td>
<td>String</td>
<td>The HTML field is a scriptable section that is parsed by Jelly and injected into the display page prior to the rest of the calendar. It can be used to pass in variables from the server and define extra fields are necessary.</td>
</tr>
<tr>
<td>Client script</td>
<td>String</td>
<td>The client script is a scriptable section that allows for configuring options of the schedule page display. The API is different depending on the schedule page view type and is discussed below.</td>
</tr>
<tr>
<td>Server AJAX processor</td>
<td>String</td>
<td>The Server AJAX processor is specific to calendar type schedule pages that is used to return a set of schedule items and spans to be displayed.</td>
</tr>
</tbody>
</table>

**Calendar views from the schedule pages**

A "URL from arguments" module, a field decoration (dictionary attribute "ref_contributions"), or a UI Action linking to "show_schedule.do" is used to invoke the Schedule page.

For example, the On-call Calendars module generates the calendar from the following URL:

```
show_schedule.do?
sysparm_type=roster&sysparm_include_view=monthly,weekly,daily,oldtimeline
```
• The URL component `sysparm_type=roster` determines which Schedule Page is used to generate the Calendar Views.
• The URL component `sysparm_include_view=monthly,weekly,daily,oldtimeline` determines which Calendar Views are available.
• The URL component `sysparm_zoom=daily` determines which Calendar View appears by default.

Timespan label events

The SchedulePage may include JavaScript functions in the Client Script section to handle events triggered by user interaction with the displayed timespans and their labels.

This is done by observing events via the CustomEvent object. For example:

```
CustomEvent.observe ( "timespanclicked" , timespanClicked )
```

In this example, your client script would include a function named `timespanClicked` to correspond to the one named in the code above.

```
function timespanClicked (event , element , scheduleItem ) { alert ( "You clicked ID=" + itemID  + " which is " + element ) ; }
```

The arguments passed to the functions are the event, the HTML element for the timespan, and the GwtScheduleItem of the timespan as given by your Schedule Page’s Server AJAX Processor code.

Timespan events that can be observed are:
• `timespan.clicked`
• `timespan.contextmenu`
• `timespan.dblclicked`
• `timespan.mouseover`
• `timespan.mouseout`

Timespan label events that can be observed are:
• `timelabel.clicked`
• `timelabel.contextmenu`
• `timelabel.dblclicked`
• `timelabel.mouseover`
• `timelabel.mouseout`

View Properties:

Specify view properties to configure aspects of the view’s appearance and behavior:
• `timespan_height`: Set to a number of pixels to customize the height of timespans on the view.
• `timespan_between`: Set to a number of pixels to customize the amount of space between timespans.
• `drag_scroll`: Set to false to prevent scrolling by dragging the timeline background left or right.

Example: Specify four timespans 4 pixels tall with 16 pixels between spans and do not allow drag scrolling.

```
schedulePage.setViewProperty("timespan_height", 4);
schedulePage.setViewProperty("timespan_between", 16);
schedulePage.setViewProperty("drag_scroll", "false");
```
Examples

There are several examples of Schedule Pages used by these plugins:

- The Group On-Call Rotation plugin contains an example of an on-call calendar.
- The Project Management v2 plugin contains the **Project Resource Timeline** Schedule Page.

Client transaction timings

The Client Transaction Timings plugin enhances the system logs by providing more information on the durations of transactions between the client and the server.

By providing information on how time was spent during the transaction, performance issues can be tracked down to the source by seeing where the time is being consumed.

This plugin requires the **Response Time Indicator** to be enabled, and collects information from the following browsers:

- Firefox
- Internet Explorer
- Chrome

Client Transactions Information

Installing the plugin adds the module Client Transactions to the System Logs application, which provides a list of every logged transaction between client and server within the last day. The following information is tracked:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>The moment the transaction was recorded.</td>
</tr>
<tr>
<td>Response Time</td>
<td>The number of ms spent by the server in fulfilling the transaction.</td>
</tr>
<tr>
<td>Business Rule Time</td>
<td>The number of ms spent by business rules triggered by the transaction.</td>
</tr>
<tr>
<td>SQL Time</td>
<td>The number of ms spent by the SQL database.</td>
</tr>
<tr>
<td>Client Response Time</td>
<td>((\text{Load_completion_time}) - (\text{start_time})). It is inclusive of server time.</td>
</tr>
<tr>
<td>Client Network Time</td>
<td>The number of ms spent by the network the client is connecting through.</td>
</tr>
<tr>
<td>Browser Time</td>
<td>The number of ms spent by the browser during the transaction.</td>
</tr>
<tr>
<td>Client Script Time</td>
<td>The number of ms spent executing client scripts</td>
</tr>
<tr>
<td>UI Policy Time</td>
<td>The number of ms spent executing ui policy</td>
</tr>
<tr>
<td>Type</td>
<td>Type of transaction (one of Form, List, Other)</td>
</tr>
<tr>
<td>Table</td>
<td>The table that was displayed e.g. incident, change_request</td>
</tr>
</tbody>
</table>
Client Detailed Information

A more detailed breakdown of the client timings for all Form rendering (but not list rendering) is also tracked. To see details, drill into a particular client transaction record and observe the related list at the base of the screen.

### Table 250: Client Detailed Information

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>The view for this form/list</td>
</tr>
</tbody>
</table>

Components installed with Client Transaction Timings

The Client Transaction Timings plugin installs several components.

**Database Table Structure**

The plugin adds the table syslog_client_transaction.

**Properties**

The property glide.client.track_transaction_timings enables and disables the plugin.

**Scripts**

The plugin relies on the new script include AJAXClientTiming. This is the script that gathers the information required and populates them on the syslog_client_transaction table.

**Dependencies**

This plugin does not require any other plugins, but will not gather information unless the Response Time Indicator is enabled. It will also gather information only from IE and Firefox browsers.

**Impact**

The plugin has little impact on systems, and can be easily disabled.
Activating the Plugin

To activate the plugin, navigate to System Definition > Plugins and activate the plugin.

*Note:* New instances have the plugin activated by default.

Disabling the Plugin

Although plugins cannot be removed, the functionality can be disabled:

1. Enter `sys_properties.list` in the application navigator filter.
2. Locate the property named `glide.client.track_transaction_timings`.
3. Set the property value to `false`.

The functionality can be enabled again by setting the property value to `true`.

Timing values

The following diagram illustrates the timing increments for rendering a page:

![Client Transaction Timing Diagram](image)

*Figure 331: Client Transaction Timing*

The variables in this diagram are defined as follows:
### Table 251: Timing Values

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_time</td>
<td>The date and time the user requests a page (the user clicks on a link). This value is set by hooking into the beforeunload event of the previous page. The beforeunload event is not properly supported by WebKit browsers, which is why the client timings are not supported on Safari or Chrome.</td>
</tr>
<tr>
<td>load_time</td>
<td>The date and time that the current page starts loading in the browser. This value is set by an inline javascript that runs as the first script in the HTML body.</td>
</tr>
<tr>
<td>server_time</td>
<td>The time in ms spent by the server processing the transaction. The server reports this value to the client.</td>
</tr>
<tr>
<td>load_completion_time</td>
<td>The date and time that the page is fully rendered in the browser. This operation is performed as the last script on the page and identifies the time the page completed loading.</td>
</tr>
</tbody>
</table>

The following times are reported by the client in the syslog_client_transaction table:

### Table 252: Timing Values

<table>
<thead>
<tr>
<th>Label</th>
<th>Element</th>
<th>Description</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time</td>
<td>client_response_time</td>
<td>Calculates the overall time to deliver the page by subtracting the time the user requests the page from the time the page is fully rendered in the browser.</td>
<td>load_completion_time - start_time</td>
</tr>
<tr>
<td>Server Time</td>
<td>client_server_time</td>
<td>This is the time the server takes to process the transaction.</td>
<td>server_time</td>
</tr>
<tr>
<td>Network Time</td>
<td>client_network_time</td>
<td>Calculates the time the network takes to process the request by subtracting the time of the user’s request from the time the page starts loading in the browser, and then subtracting the server processing time.</td>
<td>load_time - start_time - server_time</td>
</tr>
<tr>
<td>Label</td>
<td>Element</td>
<td>Description</td>
<td>Calculation</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Browser Time</td>
<td>browser_time</td>
<td>Calculates the time the browser takes to deliver the page by subtracting the time the page is fully rendered from the time the page starts loading in the browser.</td>
<td>load_completion_time - load_time</td>
</tr>
</tbody>
</table>

Define a relative duration

From the left navigation pane, select **System Scheduler > Relative Durations**.

Out of the box there are four Relative Durations in the system:

- 2 business days by 4pm
- 3 business days by 4pm
- Next business day by 4pm
- End of next business day

**Note:** Business day durations are defined by the business schedule used. If no schedule is defined, the durations are 24 X 7 by default.

We will start by looking at the End of next business day Relative Duration. From the Relative Durations list, select the **End of next business day** Relative Duration. The variable days is set to one, because we want the result of this calculation to land one day in the future. The rest of the script is as in the screenshot below. If desired, you can customize the time at which you want the Relative Duration to end (currently set to 5pm).

There is one more important Relative Duration design aspect that is used by the other three out-of-box Relative Durations. To illustrate this design we will look at **2 business days by 4pm**.

As you can see in the image below, within the script there is an if-statement. This if-statement is checking to see if the calculated time is after 10am. If it is, then an additional day is added to the calculation. Hence the description of **2 business days by 4pm if before 10am**.
Note that “End of the business day” has nothing to do with the associated Schedule. The end time of 17:00 is hardcoded into this Relative Duration script. If you want the time to be different than the out of the box 5pm, you must change it in the script.

Use a relative duration

Relative durations provide a scripted way of working out duration time for SLAs.

When you define an SLA, you can set the **Duration type** to be a relative duration.

When using relative durations, you can also use the **Relative duration works on** field to select to calculate either against the SLA record, or the task record that the SLA record is used for.

This example demonstrates how a relative duration of End of next business day works.

1. Create an SLA that has a relative duration of **End of next business day**.
2. Complete the rest of the fields of this SLA with the values as shown below, also setting a Schedule and a Timezone if you want.

3. To show how this Relative Duration works, create a new incident.

Notice that the SLA is started for this incident. If you look at the **Planned End Date** field you will notice that the date is the next business day at 5pm (See image below).
Note: Pause conditions are not compatible with Relative Durations.

Time display
There are three interfaces that display record information over time: reports, timeline pages, schedule pages.
Reports

Figure 332: Report example

Reporting allows users to generate charts from data within the platform in a variety of formats.

Any report type can report in time information, but the following report types are particularly suited to reporting on time information. See Reporting for more information.

- Calendars
- Control Charts
- Trend Charts
- Trendbox Charts
Timeline pages example

Timeline pages allow for easy definition of linear timelines from records with time information.
Event registration

After you create a new event and a business rule that uses the event, you must register it. Registration lets other parts of the system, such as Email Notifications and script actions, see the event in their list of available events and react to the event when it occurs.

Register an event

1. To register an event, browse to System Policy > Events > Registry, and then click New.
2. Complete the Event Registration form as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of your new event.</td>
</tr>
</tbody>
</table>
Introduction to time-related functionality

There are a number of different functionalities that allow for tracking time and leveraging that information across applications.

**Time Fields**

Time can be stored in different ways in records. Understanding how time is stored underlies use of all of the time functions.

For more information, see: *Use Date and Time Fields.*
Time Zones

All times are stored in the platform in Universal Coordinated Time. They are displayed globally based on the system time zone, but are displayed to users in their local time zone according to user settings. Time zone information is important to keep track of to avoid calculation errors or confusing outputs.

All dates and times gathered through web services display in GMT. Data inserted using a web service uses the active user’s time zone or the system time zone if the active user does not have a time zone specified.

For more information, see *Time zones* on page 1053.

Schedules

Schedules are rules which include or exclude ranges of time for certain time-related functionality. For example, schedules can restrict SLAs to only count time during business hours.

For more information, see *Event scheduling* on page 1016.

Displaying Time

There are a number of useful user interfaces that help represent time visually.

For more information, see *Displaying Time*.

Viewing Logs

Logs within the system provide historical information as to what occurred in the instance when.

Scheduling Events

These functionalities allow other functionality to be triggered at particular times, or in response to specific events.

For more information, see *Event scheduling* on page 1016.

Timing Functionality

These functions gather information with regards to duration. They answer the question "How long?", and can time events based off of that information.

For more information, see *Timing Functionality*.

Range calculator scripts

You can specify a script include that calculates range restrictions and processes parent updates in a timeline page.

Following are three examples of script includes that help specify range restrictions.
Range Calculator Scripts

ExampleUpdateParentsRangeCalculator

Updates parent records when a child record span is moved or resized in the timeline.

```javascript
var ExampleUpdateParentsRangeCalculator = Class.create();
ExampleUpdateParentsRangeCalculator.prototype = {
initialize: function() { },
updateParents: function(id, table, startDate, endDate){
  if (table == "rm_sprint"){
    var releaseID;
    var sprint = new GlideRecord(table);
    sprint.addQuery('sys_id', id);
    sprint.query();
    if (sprint.next())
      releaseID = sprint.release + ";";
    if (releaseID) {
      var gr = new GlideRecord("rm_release_scrum");
      gr.addQuery("sys_id", releaseID);
      gr.query();
      if (gr.next()) {
        if (startDate && startDate < this.getTimeMs(gr.start_date))
          gr.start_date = this.getTimeObject(startDate);
        if (endDate && endDate > this.getTimeMs(gr.end_date))
          gr.end_date = this.getTimeObject(endDate);
        gr.update();
      }
    }
  }
},
getMinRangeDetails: function(id, table) { return [-1, -1, "", ";"]; },
getMaxRangeDetails: function(id, table) { return [-1, -1, ""]; },
getTimeMs: function(date){
  return new GlideScheduleDateTime(date).getMS(); },
getTimeObject: function(timeMS) {
  var gdt = new GlideDateTime();
  gdt.setNumericValue(timeMS);
  return gdt; },
logMessage: function(message) { gs.log(message); },
type: 'ExampleUpdateParentsRangeCalculator'
}
```

In this example, the span is identified based on the id and table from function(id, table, startDate, endDate).

ExampleMinRangeCalculator

Defines the earliest start date and the latest end date for a specified span.

```javascript
var ExampleMinRangeCalculator = Class.create();
ExampleMinRangeCalculator.prototype = {
initialize: function() { },
updateParents: function(id, table, startDate, endDate){ },
getMinRangeDetails: function(id, table){ return [-1, -1, "", ";"]; },
getMaxRangeDetails: function(id, table){ return [-1, -1, ""]; },
getTimeMs: function(date){
  return new GlideScheduleDateTime(date).getMS(); },
getTimeObject: function(timeMS) {
  var gdt = new GlideDateTime();
  gdt.setNumericValue(timeMS);
  return gdt; },
logMessage: function(message) { gs.log(message); },
type: 'ExampleMinRangeUpdateParentsRangeCalculator'
}
```
while(gr.next()){  
    var start = this.getTimeMs(gr["start_date"]);  
    var end = this.getTimeMs(gr["end_date"]);  
    var id = gr["sys_id"];  
    if (min == -1 || start <= min){  
        if (start != min)  
            minID = ";";  
        min = start;  
        minID += "," + id;  
    }  
    if (max == -1 || end >= max){  
        if (end != max)  
            maxID = ";";  
        max = end;  
        maxID += "," + id;  
    }  
    return [min, max, minID, maxID];  
}

ExampleMaxRangeCalculator  
Defines the earliest start date and the latest end date for a specified span.
logMessage: function(message){ gs.log(message); },
type: 'ExampleUpdateParentsRangeCalculator'
}

Use the following two functions to obtain the correct start and end dates in the three example script includes provided for reference.

gTimeMs: function(date){
  return new ScheduleDateTime(date).getMS();
}

gTimeObject: function(timeMS) {
  var gdt = new GlideDateTime();
  gdt.setNumericValue(timeMS);
  return gdt;
}

Set an inactivity monitor

An inactivity monitor triggers an event for a task record if the task has been inactive for a certain amount of time. If the task remains inactive, the monitor repeats at regular intervals. User updates to the task record restart the monitor. If Reset Conditions are defined for the monitor but have not been met when you update the task record, the monitor is not restarted. Inactivity monitors only apply to records on tables that extend the Task table, or the Task table itself.

When an inactivity monitor triggers, it generates an event in the form <tablename>.inactivity (for example, incident.inactivity). The inactivity monitor does not automatically specify further actions, so either an email notification or script action must be defined to drive further action.

A record's activity is only based on user updates. System updates do not count as activity.

1. Navigate to System Policy > Inactivity Monitors and click New.
2. Give the inactivity monitor a name.
3. Specify the type of record to monitor in the Table field.
4. Specify how long the inactivity monitor should wait before sending each notification in the Wait field.
5. Specify any additional conditions in the Condition field. At least one condition must be specified for the inactivity monitor to work.
6. Specify an Order if multiple inactivity monitors might have their conditions met for a given record - the one with the lowest order will be used.
7. Click Save.

Note: If conditions are changed on an inactivity monitor, the monitor stops tracking previously tracked records. An inactivity monitor does not track records that were created before the inactivity monitor, even if the record meets all other conditions.

Escalation intervals and pause conditions

Escalation Intervals and Pause Conditions are not relevant to an inactivity monitor.

The related list and field are available because the inactivity monitor table extends the table used for SLAs, but these elements are not used in any way when an inactivity monitor attaches or is triggered.

Event scheduling

There are a variety of tools available for scheduling actions or tasks to happen in the future.
There are a variety of tools available for scheduling actions or tasks to happen in the future.

**Maintenance schedules**
Changes to the CMDB can be managed through the Maintenance Schedules Plugin, which allows changes to be proposed and viewed through a timeline.

**On-call rotation**
The Group On-Call Rotation Plugin allows a schedule to be defined to determine what users are primary contacts during particular hours of the day.

**Scheduled reports**
Once reports are defined, they can be scheduled to be emailed at a specific time, or at regular intervals, using the reporting interface.

**Scheduled workflows**
Workflows provide a robust system for automating advanced multi-step processes. Workflows can be triggered by conditions, like business rules, or they can be scheduled for a particular time/recurring schedule, like scheduled jobs.

**Scheduled jobs**
Scheduled jobs are scripts which can be set to be automatically performed at a specific date and time, or on a repeating basis.

### Scheduled jobs
Scheduled Jobs are automated pieces of work that can be performed at either a particular time, or on a recurring schedule.

These kinds of tasks can be automated:
- Run and distribute a report
- Generate a record (incident, change, configuration item, etc.) from a template
- Run a business rule and do whatever the rule contains
- Scheduling at the end of the month
- Scheduling for weekdays
- Executing scheduled jobs from scripts

### Schedule a report
Scheduled reports are distributed via email. Scheduled reports created by an individual whose user account is deactivated might not display any data. To ensure that the desired data is displayed, an active user must recreate the scheduled report.

1. Navigate to **System Definition > Scheduled Jobs**.
2. Click **New**.
3. Select **Automate the generation and distribution of a report**.
4. Populate the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name to identify this scheduled job.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Report</td>
<td>A reference to the report to be scheduled. Calendar reports and Pivot Table reports are not currently supported for automatic emailing.</td>
</tr>
<tr>
<td>Users</td>
<td>The users to send the report to at the scheduled date and time. The user must have Notification set to Email on their user record to receive reports. To force users to receive emails, use the Email Addresses field.</td>
</tr>
<tr>
<td>Groups</td>
<td>The groups to send the report to at the scheduled date and time.</td>
</tr>
<tr>
<td>Email addresses</td>
<td>Any other email addresses to send the report to, separated by commas. These emails will always receive the report, even if there's a matching user record for that address that says Do not notify.</td>
</tr>
<tr>
<td>Active</td>
<td>If true, this report will be sent at the scheduled date and time.</td>
</tr>
<tr>
<td>Run</td>
<td>The type of schedule to send the report on. Choices are:</td>
</tr>
<tr>
<td></td>
<td>- Daily</td>
</tr>
<tr>
<td></td>
<td>- Weekly</td>
</tr>
<tr>
<td></td>
<td>- Monthly</td>
</tr>
<tr>
<td></td>
<td>- Periodically</td>
</tr>
<tr>
<td></td>
<td>- Once</td>
</tr>
<tr>
<td>Day</td>
<td>• If Run is Weekly, the day of the week</td>
</tr>
<tr>
<td></td>
<td>• If Run is Monthly, the day of the month.</td>
</tr>
<tr>
<td>Repeat Interval</td>
<td>If Run is Periodically, the duration between each scheduled report. The duration can be days, hours, or minutes.</td>
</tr>
<tr>
<td>Time</td>
<td>If Run is Weekly or Monthly the time of day, on a 24 hour clock.</td>
</tr>
<tr>
<td>Starting</td>
<td>The date and time of the first scheduled report.</td>
</tr>
<tr>
<td>Priority</td>
<td>The numerical priority of a scheduled job. Set essential jobs to a priority value below 100 and nonessential jobs to a priority above 100. If 70% or more of all scheduled jobs are Overdue, any jobs marked with a value above 100 do not run.</td>
</tr>
<tr>
<td>Subject</td>
<td>The subject line for the email.</td>
</tr>
<tr>
<td>Introductory Message</td>
<td>The body of the email.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Run as</td>
<td>The user creating and running the scheduled job. To have the scheduled job assigned to system behavior instead of the person creating the scheduled job, create a system or dummy user and add that user to this field.</td>
</tr>
<tr>
<td>Type</td>
<td>The file-type of the attached report. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• PDF-landscape</td>
</tr>
<tr>
<td></td>
<td>• PDF</td>
</tr>
<tr>
<td></td>
<td>• Excel</td>
</tr>
<tr>
<td></td>
<td>• CSV</td>
</tr>
<tr>
<td></td>
<td>• PNG</td>
</tr>
<tr>
<td>Include Detail</td>
<td>If checked, will include details on the records in the report.</td>
</tr>
<tr>
<td>Zip Output</td>
<td>If checked, the report will be compressed in a .zip file.</td>
</tr>
</tbody>
</table>

As a workaround for scheduling calendar and pivot table reports for email distribution, consider using the publish option. Publish creates a URL for the report and displays the address above the report form. You can create an email notification with this URL and send the link to people who need to see the report, or you can send the URL for the calendar report to a distribution list.

**Schedule a script execution**

Non-conditional and conditional scripts can be scheduled.

1. Navigate to System Definition > Scheduled Jobs.
2. Click New.
3. Select **Automatically run a script of your choosing**.
4. Populate the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name to identify this scheduled script execution</td>
</tr>
<tr>
<td>Active</td>
<td>If true, the script will be executed at the scheduled date and time.</td>
</tr>
</tbody>
</table>
### Field | Input Value
--- | ---
**Run** | The type of schedule to execute the script on. Choices are:
- Daily
- Weekly
- Monthly
- Periodically
- Once

**Day** | • If **Run** is *Weekly*, the day of the week
• If **Run** is *Monthly*, the day of the month.

**Repeat Interval** | If **Run** is Periodically, the duration between each script execution.

**Time** | If **Run** is *Weekly* or *Monthly* the time of day, on a 24 hour clock.

**Starting** | The date and time of the first scheduled script execution.

**Application** | The application that contains the script.

**Conditional** | If checked, the entity will only be executed if certain conditions are met.

**Condition** | If **Conditional** is checked, a script determines under what conditions the scheduled script is executed. The last expression of the script should evaluate to a Boolean (true/false) value.

**Run as [Optional]** | Select another user to run the script execution as. Configure the form to add this field if it is not present.

**Run this script** | The script to run at the scheduled date and time. For example, copy script logic from a business rule or call a script include.

### Example
The following is an example of a conditional script. This example runs the scheduled job only if there are active Incidents older than 30 days.

```javascript
// Only run this Scheduled Job if there are active Incidents over 30 days old
var ga = new GlideAggregate('incident');
ga.addAggregate('COUNT');
ga.addQuery('active', 'true');
ga.addQuery('sys_created_on', '<', gs.daysAgo(30));
ga.query();
ga.next();
ga.getAggregate('COUNT') !== '0'
```
Schedule the generation of an entity

Entities include changes, incidents, and CIs.

1. Navigate to **System Definition > Scheduled Jobs**.
2. Click **New**.
3. Select **Automatically generate something (a change, an incident, a ci, etc) from a template**.
4. Populate the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>If true, the entity will be generated at the scheduled date and time.</td>
</tr>
<tr>
<td>Run</td>
<td>The type of schedule to generate the entity on. Choices are:</td>
</tr>
<tr>
<td></td>
<td>- Daily</td>
</tr>
<tr>
<td></td>
<td>- Weekly</td>
</tr>
<tr>
<td></td>
<td>- Monthly</td>
</tr>
<tr>
<td></td>
<td>- Periodically</td>
</tr>
<tr>
<td></td>
<td>- Once</td>
</tr>
<tr>
<td>Day</td>
<td>- If Run is <strong>Weekly</strong>, the day of the week</td>
</tr>
<tr>
<td></td>
<td>- If Run is <strong>Monthly</strong>, the day of the month</td>
</tr>
<tr>
<td>Repeat Interval</td>
<td>If Run is <strong>Periodically</strong>, the duration between each scheduled generation.</td>
</tr>
<tr>
<td>Time</td>
<td>If Run is <strong>Weekly</strong> or <strong>Monthly</strong> the time of day, on a 24 hour clock.</td>
</tr>
<tr>
<td>Starting</td>
<td>The date and time of the first scheduled generation.</td>
</tr>
<tr>
<td>Conditional</td>
<td>If checked, the entity will only be generated if certain conditions are met.</td>
</tr>
<tr>
<td>Condition</td>
<td>If Conditional is checked, a script determines under what conditions the</td>
</tr>
<tr>
<td></td>
<td>scheduled script is executed. The last expression of the script should</td>
</tr>
<tr>
<td></td>
<td>evaluate to a Boolean (true/false) value.</td>
</tr>
<tr>
<td>Generate this</td>
<td>A reference to a template for a record.</td>
</tr>
</tbody>
</table>

Special cases in job schedules

Some special cases require care in job scheduling.
End of the month schedules

Because months have different lengths, take care when scheduling jobs for the end of the month.

• Scheduling an event for the 29th or 30th is not recommended, because the scheduled job will not be executed in months (like February) which are shorter than those dates.
• If an event is scheduled for the 31st, it will be executed on the last day of the month, even if the month is shorter.

For example, something scheduled to run on the 31st of the month will run on February 28th or February 29th in a leap year.

Weekday schedules

For scheduled scripts, use the following script to run only on weekdays:

```javascript
function checkWeekdays() {
    var now = new Date();
    var day = now.getDay();
    var result = false;
    if(day != 0 && day != 6) {
        result = true;
    }
    return result;
}
checkWeekdays();
```

Scheduled jobs from scripts

To execute a scheduled job triggered by an event, use the following script:

```javascript
//Execute a scheduled script job
var rec = new GlideRecord('sysauto_script');
rec.get('name', 'YOUR_JOB_NAME_HERE');
SncTriggerSynchronizer.executeNow(rec);
```

This script can be run using one of several tables:

• scheduled_import_set (Scheduled Import Sets)
• sysauto_script (Scheduled Script Execution)
• sysauto_template (Scheduled Template Generation)
• sysauto_report (Scheduled Report)

Note: SncTriggerSynchronizer does not provide methods to execute scheduled jobs in the future.

Running Scheduled Jobs Imported from Another Instance

To prevent unexpected data changes, the system does not automatically create Schedule Item [sys_trigger] records for Scheduled Jobs [sysauto] imported from an XML file such as an update set. To run a scheduled job imported from another instance, update the scheduled job record.
Create a scheduled job

Create a scheduled job on the Schedule Job [sysauto] table (System Definition > Scheduled Jobs).
Create all new scheduled jobs using this method. Some existing scheduled jobs are found on the Schedule Item [sys_trigger] table (System Scheduler > Scheduled Jobs). Do not create new scheduled jobs on the Schedule Item table.

To create a scheduled job:
1. Navigate to System Definition > Scheduled Jobs.
2. Click New.
3. Select the appropriate type of scheduled job.

The fields presented will depend on the type of scheduled job required.

View a schedule item

Schedule items are individual instances of a scheduled job.

To see which scheduled jobs will run today:

Navigate to System Scheduler > Today’s Scheduled Jobs.
The table displays each schedule item that will be run.

Note: It is usually inadvisable to modify the schedule items themselves. It is best to modify the scheduled jobs that contain them.

Event registry

Events are registered in the Event Log by Business Rules. These events can be used to automate other activities, such as script actions or notifications.

Events can be used to schedule actions or tasks to occur when conditions are fulfilled.

Examples:
• kb.view - an event triggered when a user views a Knowledge Base article, used to trigger the script action Knowledge View to create a Knowledge Use record every time an article is viewed.
• incident.commented - an event triggered when a user comments on an article, used by two incident commented email notifications.

Inactivity Monitors:
An inactivity monitor triggers an event if a record has not been updated for a defined length of time.

**Script Actions:**

Script actions are scripts which are triggered when an event is recorded in the log. In that way, scripts can be set to be performed whenever a particular activity occurs in the platform, rather than at a particular time (like scheduled jobs) or in response to particular conditions (like a business rule).

**Notifications:**

Events are also used to trigger Email Notifications when an event is recorded in the log.

**System scheduler**

The System Scheduler application contains two separate engines for scheduling: the Scheduled Jobs engine and the Schedules engine.

Functionality described here requires the Admin role.

**Scheduled Jobs:**

Scheduled Jobs performs any work that must be done at a specific time or on a recurring basis. The Scheduled Jobs module links to the Schedule [sys_trigger] table. Manipulating records on the Schedule table is not recommended. Use this table to view existing base system scheduled jobs.

The Scheduled Jobs module in System Definition is an admin-friendly front end for scheduling work. Use this module to create new scheduled jobs.

For more information, see [Create a Scheduled Job](#).

**Schedules:**

Schedules are rules which include or exclude time on a calendar. They are used by service levels, inactivity monitor, and group on-call rotation. For instance, a schedule can be defined to restrict service levels to only apply to weekdays during business hours, or to exclude holidays from an on-call rotation.

For more information, see [Use Schedules](#).

**Timeline pages**

Use timeline pages to track any activity bounded by two dates, such as the start and end date of a change request or the open and close date of an incident. For details on timeline navigation, filtering, and working with calendar perspectives, see [Using Timelines](#).
Features

- Make selected timelines available to users by role.
- Select perspective from daily to yearly views.
- Create dynamic labels for timeline spans.
- Configure tooltips for each span.
- Permit span dragging and resizing by users.
- Lock timelines to prevent editing.

Timelines

Timelines display a linear calendar of tasks, such as incidents and change requests, defined by their start and end dates.

Each task on the timeline is represented by a span, which is displayed as a horizontal bar and may have distinctive color coding. Each span has a label and can have tooltip text containing additional information about the task. You can also change the calendar’s perspective for a more granular view of the data.

Timeline pages

Timeline pages are created by an administrator and are user interactive.

They can be navigated and zoomed, and records can be updated from within the timeline by manipulating the spans.

Viewing Timelines

By default, only administrators have a module to access timeline records. ITIL users can only view timelines by selecting a context menu item from an incident record.
Figure 336: Timeline new

Administrator Role Users

Users with an administrator role can view timelines from the timeline page records.

1. Navigate to System UI > Timeline Pages.
2. Select a timeline record to view.
3. Click View Timeline.

Other Users

ITIL users can view timelines associated with any task record where the metrics are set to display as timeline records (by default, only Incident records display timeline metrics). ITIL users can select the
**Metrics Timeline** context menu UI action to view the Assigned to duration and the State duration metric timelines.

1. Navigate to active task record, such as an incident.
2. Right-click the banner, and select **Metrics Timeline**.

![Incident](image)

**Figure 337: Metrics timeline**

To allow non-administrators to view other timelines, create a custom module.

**Span details**
- If the left navigation pane is visible, the span label is visible in the left pane.
- If the left pane is hidden, hover the cursor over a span to view the tooltip.
Figure 338: Timeline tooltip

Span focus and zoom

Table 257: Focusing and Zooming on a Span

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macintosh</td>
<td>Command + Click</td>
</tr>
<tr>
<td>Windows</td>
<td>Control + Click</td>
</tr>
</tbody>
</table>

Spans edits

If the timeline has been configured for editing (horizontal moving, start and end date dragging), use the mouse to adjust the spans to suit your needs.

Dragging the span and changing the start or end dates updates the parent record.

Move the start or end dates to change the duration of the task.

Figure 339: Timeline drag start

Move the entire span horizontally to adjust the start and end dates but not the duration.
High Priority Incidents

Move the span horizontally

<table>
<thead>
<tr>
<th>INCO0000041, My cubicle</th>
<th>INCO0000005, CPU load high for over 10 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCO0000014, missing my home directory</td>
<td>INCO0000044, Cant log into server</td>
</tr>
<tr>
<td>INCO0000027, please remove this host</td>
<td>INCO0000016, Rain is leaking on main DNS server</td>
</tr>
</tbody>
</table>

Figure 340: Timeline drag span

Changes in perspective
Use the range selectors at the top of the timeline to change the perspective.

The increments go from one day to one year. To limit the timeline to an increment between the start date of the first span and the end date of the last span, click Max. Use the starting and ending calendar fields to select the timeline perspective. These fields control the same perspective as the slider at the bottom of the timeline. The green, vertical line indicates the current date and time, and sweeps across the timeline automatically.

Figure 341: Timeline perspective bar

The increments go from one day to one year. To limit the timeline to an increment between the start date of the first span and the end date of the last span, click Max. Use the starting and ending calendar fields to select the timeline perspective. These fields control the same perspective as the slider at the bottom of the timeline. The green, vertical line indicates the current date and time, and sweeps across the timeline automatically.

Figure 342: Timeline perspective bar

The pink slider at the bottom of the timeline offers another way to change the perspective. Move the slider from right to left to view all the spans on a long timeline. Adjust the end points of the slider to make arbitrary changes to the magnification. A narrow slider zooms in on the spans and provides a more detailed view of complex timelines. A wide slider pulls the view out and makes more of the timeline visible on the screen.
Create a timeline page

1. Navigate to **System UI > Timeline Pages**.
2. Click **New**.
3. Fill in the form as described in the table and click **Submit**.

**Table 258: Creating A Timeline Page**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name that describes the function of this timeline. For example, High Priority Change Requests.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the name of the table associated with this timeline, such as Change Request [change_request].</td>
</tr>
<tr>
<td>Start date field</td>
<td>Select a time-related field from the specified table to use as the start date for the timeline. The timeline begins with the span for the record with the earliest start date from this field, after the filter and sort order are applied. For example, you can select Updated as the start date field and start the span for each active change request on the date it was updated to a high priority.</td>
</tr>
<tr>
<td>End date field</td>
<td>Select a time-related field from the specified table to use as the end date for the timeline. The timeline ends with the span for the record with the latest date from this field, after the filter and sort order are applied. For example, you can select Closed as the end date field and display all high priority change requests by the date on which they were closed.</td>
</tr>
</tbody>
</table>

**Display Options**

<p>| Show grid lines? | Select this check box to show horizontal background shading to highlight alternate spans. |
| Show left pane?  | Select this check box to show label text in a pane on the left of the timeline. The text that appears in this pane is defined in Span text fields. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show summary pane</td>
<td>Select this check box to show the pink, perspective slider at the bottom of the timeline. Move the slider from right to left to scroll across the chart. Adjust the end points of the slider to change the magnification. A narrow slider zooms in on the spans and provides a more detailed view. A wide slider pulls the view out and makes more of the timeline visible on the screen.</td>
</tr>
<tr>
<td>Auto refresh</td>
<td>Select an automatic refresh interval or disable automatic refresh. When auto refresh is disabled, the timeline adjusts only when the browser is manually refreshed or when a start or end date field is updated in a record.</td>
</tr>
<tr>
<td>CSS span color</td>
<td>Enter a custom span color using any CSS color format, such as RGB or hexadecimal. If this field is blank, the default span color, light blue, is used.</td>
</tr>
<tr>
<td>Show span text</td>
<td>Select this check box to display the content of the Span text fields as labels below each span.</td>
</tr>
<tr>
<td>Span text fields</td>
<td>Select fields from the specified table to have those values appear as span labels. For example, you might select Number and Short description. The span labels also appear in the left pane if the left pane is visible.</td>
</tr>
<tr>
<td>Show tooltips</td>
<td>Select this check box to display tooltips when the cursor rests on a span.</td>
</tr>
<tr>
<td>Tooltip text fields</td>
<td>Select from the specified table the fields whose values appear as tooltips. For example, you might select Category, Assigned to, and Due date.</td>
</tr>
<tr>
<td>Filtering and Sorting</td>
<td>Create a condition to filter the results that appear in the timeline. For example, a condition that displays only active, high priority incidents. This field has a Condition Count Widget to preview what records will be returned by this condition set.</td>
</tr>
<tr>
<td>Perform custom sort?</td>
<td>Select this check box to enable custom sorting. Configure the sort order by selecting fields in the Sort by and Sort by order fields.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Sort by</td>
<td>Select any field in the list for sorting the spans in the timeline. Common practice is to select either the Start date field or the End date field as the sorting field. If you select a different sorting field, also include that field in the list of Tooltip text fields to give users a way of discovering the sort criteria.</td>
</tr>
<tr>
<td>Sort by order</td>
<td>Select the sort order for the sorting fields selected.</td>
</tr>
<tr>
<td><strong>Interactive Options</strong></td>
<td></td>
</tr>
<tr>
<td>Allow horizontal moving?</td>
<td>Select this check box to permit users to drag timeline spans horizontally. Dragging changes the start and end dates and updates the record.</td>
</tr>
<tr>
<td>Allow start time dragging?</td>
<td>Select this check box to permit users to update the record by dragging the start time of a span.</td>
</tr>
<tr>
<td>Allow end time dragging?</td>
<td>Select this check box to permit users to update the record by dragging the end time of a span.</td>
</tr>
<tr>
<td>Range calculator</td>
<td>Specify a script include that calculates range restrictions and processes parent updates, if appropriate.</td>
</tr>
</tbody>
</table>

A completed Timeline Page record looks like this.
Customize the timeline page span style

The Timeline Page Span Style related list allows you to define conditional span styles.

1. Navigate to **System UI > Timeline Pages**.
2. Open the timeline page for which you want to define the span style.
3. Go to the Timeline Page Span Styles related list and click **New**.
4. Fill in the form as described in the table and click **Submit**.

The Timeline Page Span Style form provides the following fields.
Table 259: Customizing the Timeline Page Span Styles

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>Create a condition to filter the results that appear in the timeline.</td>
</tr>
<tr>
<td>Label color</td>
<td>Select the color for the text under each span.</td>
</tr>
<tr>
<td>Label decoration</td>
<td>Select a character style for the text under each span: Bold, Italic, Underline, or Line-through.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to determine the sequence in which the style conditions are evaluated. Style conditions with a lower order are evaluated first.</td>
</tr>
<tr>
<td>Span color</td>
<td>Select the color of each span.</td>
</tr>
<tr>
<td>Timeline page</td>
<td>Select the timeline page to which the span style applies. By default, the span applies to the current timeline.</td>
</tr>
</tbody>
</table>

Display a metric as a timeline

Administrators can allow users to display any metric on a timeline by activating the Timeline Metrics UI action.

1. Navigate to Metrics > Definitions.
2. Select the metric you want to display on a timeline. For example, Problem State Duration.
3. Select the Timeline checkbox.
4. Click Update.

The UI action is available on the same table as the metric. By default, only Incident metrics are available.
Make a timeline visible to a selected user

Make selected timelines available to users by creating a custom module within an application and defining the roles that can access it.

To permit these users to update task records directly from the timeline, configure the timeline to allow span dragging.

**Note:** Timelines delivered by a custom module are not entirely dynamic. The left pane, the summary pane, the auto-refresh feature, and the grid lines are not dynamic and do not reflect changes made to the timeline record after the module link is created. However, the data represented by the spans, the labels, and tooltips display all updates in the custom module.

To create a timeline page module:

1. Right-click an application (such as Incident) in the navigation pane and select **Edit application**.
2. In the application Modules related list, click **New**.
3. **Configure** the Module form to add the Timeline page field.
4. Fill in the form as described in the table and click Submit.

The Module form provides the following fields.

Table 260: Making Timelines Visible to Selected Users

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the module as it will appear in the navigation pane. For example, you might use Planning Timeline.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to determine the sequence in which this condition will be evaluated if more than one matching condition exists. Conditions with a lower order are evaluated first.</td>
</tr>
<tr>
<td>Application</td>
<td>Select the application for the new module.</td>
</tr>
<tr>
<td>Hint</td>
<td>Enter a brief description of the module that appears when the user places the cursor over the module name. For example, you might enter Weekly view of high priority changes.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to enable the module for the roles defined. Clear this check box to disable the module for all users.</td>
</tr>
<tr>
<td>Image</td>
<td>Select an appropriate icon to appear with the module name.</td>
</tr>
<tr>
<td>Link type</td>
<td>Select Timeline Page. When this link is selected, the Timeline Page field appears.</td>
</tr>
<tr>
<td>Timeline Page</td>
<td>Select the timeline page you want to appear in this module. For example, for the Change application select a change-related timeline, for the Incident application select an incident related timeline.</td>
</tr>
<tr>
<td>Roles</td>
<td>Select the roles that can access this module.</td>
</tr>
</tbody>
</table>

The completed module form looks like this.
Timeline sub item

Use the Timeline Sub Items related list to define child spans for the timeline, based on records in a table that references the parent timeline's table.

This can be used to generate a hierarchical relationship starting from a timeline page to any number of levels. For example, if there is a timeline page for a release, a sub item might be sprints, and a sprint might have stories as a sub item.

To create a new sub item:

1. Navigate to System UI > Timeline Pages.
2. Open the timeline page for which you want to add a sub item.
3. Go to the Timeline Sub Items related list and click New.
4. Fill in the form as described in the table and click Submit.

The Timeline Sub Item form provides the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent</td>
<td>[Read-only] Identifies the parent of the timeline sub item.</td>
</tr>
<tr>
<td>Name</td>
<td>Enter a unique name that describes the function of this timeline. For example, Sprints for High Priority Changes.</td>
</tr>
</tbody>
</table>

Table 261: Timeline Sub Items
<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select the table called by this timeline. The selected table must have at least one reference field to the table selected for the parent timeline page. For example, if the parent timeline page uses the Scrum Release [rm_release_scrum] table, you might choose the Sprint [rm_sprint] table for a timeline sub item.</td>
</tr>
<tr>
<td>Start date field</td>
<td>Select a time-related field from the specified table to use as the start date for the timeline. For example, Planned start date.</td>
</tr>
<tr>
<td>End date field</td>
<td>Select a time-related field from the specified table to use as the end date for the timeline. For example, Planned end date.</td>
</tr>
<tr>
<td>Parent Reference Column</td>
<td>Select a reference field on which to base the timeline connection between the sub item records and the parent records. If multiple reference fields are available, choose the reference field that forms part of the hierarchy to be modeled by this timeline. If this list is blank, the sub item table contains no reference fields to the parent table. In this case, you must choose a different table for the sub item. ServiceNow uses the parent reference column to determine which records are displayed at each level of the timeline.</td>
</tr>
<tr>
<td>Display Options</td>
<td></td>
</tr>
<tr>
<td>CSS span color</td>
<td>Enter a custom span color using any CSS color format, such as RGB or hexadecimal. If this field is blank, the default span color, light blue, is used.</td>
</tr>
<tr>
<td>Span text fields</td>
<td>Select fields from the specified table to have those values appear as span labels. For example, you might select Number and Short description. The span labels also appear in the left pane if the timeline displays the left pane.</td>
</tr>
<tr>
<td>Tooltip text fields</td>
<td>Select fields from the specified table to have those values appear as tooltips. For example, you might select Category, Assigned to, and Due date.</td>
</tr>
<tr>
<td>Filtering and Sorting</td>
<td></td>
</tr>
<tr>
<td>Condition builder</td>
<td>Create a condition to filter the results that appear in the sub item. For example, you might create a condition that displays only active, high priority incidents.</td>
</tr>
<tr>
<td>Field</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Perform custom sort</td>
<td>Select this check box to enable custom sorting. Configure the sort order by selecting fields in the Sort by and Sort by order fields.</td>
</tr>
<tr>
<td>Sort by</td>
<td>Select any field in the list for sorting the spans in the timeline. Common practice is to select either the Start date field or the End date field as the sorting field. If you select a different sorting field, also include that field in the list of Tooltip text fields to give users a way of discovering the sort criteria.</td>
</tr>
<tr>
<td>Sort by order</td>
<td>Select the sort order for the sorting fields selected.</td>
</tr>
<tr>
<td>Interactive options</td>
<td></td>
</tr>
<tr>
<td>Allow horizontal moving?</td>
<td>Select this check box to permit users to drag timeline spans horizontally. Dragging changes the start and end dates and updates the record.</td>
</tr>
<tr>
<td>Allow start time dragging?</td>
<td>Select this check box to permit users to update the record by dragging the start time of a span.</td>
</tr>
<tr>
<td>Allow end time dragging?</td>
<td>Select this check box to permit users to update the record by dragging the end time of a span.</td>
</tr>
<tr>
<td>Restriction</td>
<td>Specify the behavior when dragging a child span (available only if no Range calculator was specified for the parent timeline page).</td>
</tr>
<tr>
<td></td>
<td>• None: No restriction is in place.</td>
</tr>
<tr>
<td></td>
<td>• Restrict by parent: Child span can be moved only within the time frame defined by the parent span.</td>
</tr>
<tr>
<td></td>
<td>• Update parent: Parent span is updated when the child span is moved outside the time frame defined by the parent span.</td>
</tr>
</tbody>
</table>

Timing functionality

Timing functionality are tools that exist to answer the question "How Long?"

Metric definitions

Defined metrics can track how long an audited field holds a certain value.

For instance, a metric can track how long an incident is assigned to an individual, or how long an incident is in the state Active.
SLAs

Service Level Agreements time how long a task meets a certain condition, and is primarily used to ensure that tasks are handled within a pre-determined time limit.

SLAs define the following conditions:

- Start Conditions
- Pause Conditions
- Stop Conditions

Once a task meets the Start Conditions, the SLA will time how long the task remains in that condition (unless it meets Pause Conditions). The timer will end if the Stop Conditions are met. If the time-limit is passed, the SLA will be marked breached.

Notifications can be driven off of the SLA to warn interested parties as the time limit approaches.

Time worked fields

The Task [task] table provides a time-tracking field called Time worked. This field measures how long a record has been viewed in order to measure work time on a ticket.

Figure 344: Time Worked

Any table that extends Task can use this field. To add the field, configure the form.

As the record is viewed, the timer counts upward. To pause the timer, click the stop icon (嘴巴).

To resume the timer, click the start icon (嘴巴).

When the task is saved, the amount of new time in the timer is used to generate a record on the Time Worked [task_time_worked] table. This table can be viewed as a related list on the task form.

By default, the time displayed in the Time worked field displays a cumulative value stored in the task record. If you modify a Time Worked record, the changes will not be reflected in the task timer.

You can set the property com.snc.time_worked.update_task_timer to enable updating of the task timer value based on changes to the time worked records. This is accomplished through the Update task timer business rule.

Schedules

Schedules are rules that include or exclude time for various actions or tasks.

Use schedules to specify when service level agreements or inactivity monitors are active, or to specify when on-call rotations should take effect. For example, if a service level agreement is set to an 8-5 Weekdays schedule, the SLA only counts time during those hours.

Default schedules

Default schedules are available in a base system.
Table 262: Default Schedules

<table>
<thead>
<tr>
<th>Name</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-5 weekdays</td>
<td>Repeats every week on weekdays Monday through Friday.</td>
</tr>
<tr>
<td>8-5 weekdays excluding holidays</td>
<td>Repeats every week on weekdays Monday through Friday. Includes the child schedule U.S. Holidays.</td>
</tr>
<tr>
<td>Name</td>
<td>Schedule</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application</td>
<td>Repeats every week on Sunday.</td>
</tr>
<tr>
<td>Application FLX</td>
<td>Repeats every week on Sunday.</td>
</tr>
<tr>
<td>Blackout Wednesdays (GMT)</td>
<td>Repeats every week on Wednesday.</td>
</tr>
<tr>
<td>Database Server FLX</td>
<td>Repeats every week on Saturday.</td>
</tr>
<tr>
<td>Default MS Project</td>
<td>Repeats every week on weekdays Monday through Friday from 8:00am to noon and 1:00-5:00pm. Default schedule for the Project Management application.</td>
</tr>
<tr>
<td>Global Infrastructure</td>
<td>Repeats every week on Saturday.</td>
</tr>
<tr>
<td>MySQL Database Service</td>
<td>Repeats every week on Saturday.</td>
</tr>
<tr>
<td>Network</td>
<td>Repeats every week on Saturday.</td>
</tr>
<tr>
<td>Project Management Schedule</td>
<td>Repeats every week on weekdays Monday through Friday from 8:00am to noon and 1:00-5:00pm. Default schedule for the Project Management application.</td>
</tr>
<tr>
<td>Resource Management Schedule</td>
<td>Repeats every week on weekdays Monday through Friday from 8:00am to noon and 1:00-5:00pm. Default schedule for the Resource Management application.</td>
</tr>
<tr>
<td>Server</td>
<td>Repeats every week on Saturday.</td>
</tr>
<tr>
<td>Servers San Diego</td>
<td>Repeats every week on Wednesday.</td>
</tr>
<tr>
<td>Software Blackout</td>
<td>Repeats every week on Wednesday.</td>
</tr>
<tr>
<td>WebServer FLX</td>
<td>Repeats every week on Sunday.</td>
</tr>
<tr>
<td>Weekends</td>
<td>Repeats every week on Saturday for two days.</td>
</tr>
</tbody>
</table>

**Holidays**

Each individual holiday can be defined as a schedule entry to create exceptions to existing schedules. For instance, if an SLA requires an incident be resolved within three business days excluding Christmas, create a schedule entry for Christmas to ensure that SLAs do not count Christmas when calculating elapsed time, even if it falls within the work week.

Because schedules can be included in other schedules through a parent-child relationship, it is also possible to create a holiday schedule and include it in other schedules to keep holidays consistent.

The following example shows a holiday schedule.
## Schedule - U.S. Holidays

### Name:
- **U.S. Holidays**

### Time zone:
- **Floating**

### Parent:
- **Parent**

### Type:
- **Type**

### Description:
- Sample set of holidays recognized in the United States

---

## Related Links

**Show Schedule**

### Schedule Entries (12)

<table>
<thead>
<tr>
<th>Name</th>
<th>Repeats</th>
<th>Repeat every</th>
<th>Start date time</th>
<th>End date time</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christmas Day</td>
<td>Every year on Dec 25</td>
<td>1</td>
<td>12-25-2012 00:00:00</td>
<td>12-25-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Christmas Eve</td>
<td>Every year on Dec 24</td>
<td>1</td>
<td>12-24-2012 00:00:00</td>
<td>12-24-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Columbus Day</td>
<td>Every year on 2nd Mon of Oct</td>
<td>1</td>
<td>10-09-2012 00:00:00</td>
<td>10-09-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Independence Day</td>
<td>Every year on July 4</td>
<td>1</td>
<td>07-04-2012 00:00:00</td>
<td>07-04-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Labor Day</td>
<td>Every year on 1st Mon of Sep</td>
<td>1</td>
<td>09-01-2012 00:00:00</td>
<td>09-01-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Martin Luther King, Jr. Day</td>
<td>Every year on 3rd Mon of Jan</td>
<td>1</td>
<td>01-16-2012 00:00:00</td>
<td>01-16-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Memorial Day</td>
<td>Every year on last Mon of May</td>
<td>1</td>
<td>05-28-2012 00:00:00</td>
<td>05-28-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>New Year's Day</td>
<td>Every year on Jan 1</td>
<td>1</td>
<td>01-01-2012 00:00:00</td>
<td>01-01-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>New Year's Eve</td>
<td>Every year on Dec 31</td>
<td>1</td>
<td>12-31-2012 00:00:00</td>
<td>12-31-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Presidents' Day</td>
<td>Every year on 3rd Mon of Feb</td>
<td>1</td>
<td>02-20-2012 00:00:00</td>
<td>02-20-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Thanksgiving Day</td>
<td>Every year on 4th Thu of Nov</td>
<td>1</td>
<td>11-22-2012 00:00:00</td>
<td>11-22-2012 23:59:59</td>
<td>Excluded</td>
</tr>
<tr>
<td>Veterans Day</td>
<td>Every year on Nov 11</td>
<td>1</td>
<td>11-11-2012 00:00:00</td>
<td>11-11-2012 23:59:59</td>
<td>Excluded</td>
</tr>
</tbody>
</table>
The following example shows a schedule that includes the holiday schedule shown above.

**Figure 346: Child Schedule**

Create a holiday schedule for multiple regions

You can create holiday schedules for multiple regions that follow the same work schedule but have different holidays.
The following method supports multiple regions that all follow the same work schedule (for example, an 8-5 weekdays schedule) but have different holiday schedules.

1. Create a holiday schedule for each region. For example, U.S. Holidays, British Holidays, and Australian Holidays.

2. Add the work schedule as a child schedule to each region’s holiday schedule.

This method requires making \(<\text{number of schedules}\> + 1\) total schedules. If you instead make the regional holiday schedule a child schedule of the work hours schedule, you will need to create a separate work hours schedule for each region. The total number of schedules in this case is \(<\text{number of schedules}\> \times 2\) schedules.

Parent and child schedules

Schedules can have one of two parent-child relationship with other schedules.

- **Parent field:** When a schedule record lists a value for the **Parent** field, schedule entries from the parent schedule apply to both the parent schedule and the child schedule. By default, there are no sample schedules that use the **Parent** field.
- **Child schedule:** When a schedule record has one or more child schedules in the **Child Schedules** related list, schedule entries from the child schedule apply to the containing schedule. By default, there are several sample schedules that use child schedules. For example, see the **8-5 weekdays excluding holidays** schedule that includes the **U.S. Holidays** schedule.

Parent and child schedules cannot contain conflicting schedule entry types. For example, a schedule containing maintenance schedule entries cannot also contain blackout schedule entries. Nor can a maintenance schedule have a child schedule containing blackout schedule entries.

Parent schedules are not valid if they are only exclusionary. They must have at least one entry that is not of type **Excluded**.

**Note:** The **Show Schedule** related link shows schedule entries from the current schedule and the child schedule record. For example, when showing the **8-5 weekdays excluding holidays** schedule, holidays are also shown as excluded because the holiday schedule is a child schedule.

Schedule entry fields

The **Schedule Entries** related list contains the definitions of the time periods you want to include in or exclude from the schedule.

**Note:** A schedule entry can only be associated with one schedule.

The Schedule Entry form uses these fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the schedule entry.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>Enter a label that describes the purpose of the schedule. The system also uses the schedule type to determine how to process certain schedules.</td>
</tr>
<tr>
<td></td>
<td>• Excluded: excludes time periods from SLA counts.</td>
</tr>
<tr>
<td></td>
<td>• Maintenance: specifies time periods where change management activities are allowed. A schedule containing maintenance schedule entries cannot also contain blackout schedule entries.</td>
</tr>
<tr>
<td></td>
<td>• Blackout: excludes time periods from change management schedules. A schedule containing blackout schedule entries cannot also contain maintenance schedule entries.</td>
</tr>
<tr>
<td>Show As</td>
<td>Select an option to indicate how the schedule entry should be displayed in calendar applications and how it should interact with other schedule entries.</td>
</tr>
<tr>
<td>When</td>
<td>Enter the date and time to which the schedule entry applies. If the schedule entry applies to a full 24-hour day, select the All day check box.</td>
</tr>
<tr>
<td>Repeats</td>
<td>Select a repetition interval for the schedule entry, if any. If you select a repetition interval, ServiceNow displays other fields to further specify the repeat interval.</td>
</tr>
<tr>
<td>Repeat every</td>
<td>Select how often the schedule repeats daily, weekly, monthly, or yearly. This field is only visible when the Repeats field has a value of Daily, Weekly, Monthly, or Yearly.</td>
</tr>
<tr>
<td>Repeat on</td>
<td>Select the days of the week a weekly schedule repeats on. This field is only visible when the Repeats field has a value of Weekly.</td>
</tr>
<tr>
<td>Monthly type</td>
<td>Select how a monthly schedule repeats. This field is only visible when the Repeats field has a value of Monthly. Monthly repeat options include:</td>
</tr>
<tr>
<td></td>
<td>• Repeat on a specific day of the month</td>
</tr>
<tr>
<td></td>
<td>• Repeat on a specific day in a specific week of the month</td>
</tr>
<tr>
<td></td>
<td>• Repeat on the last day of the month</td>
</tr>
<tr>
<td></td>
<td>• Repeat on a specific week day in the last week of the month</td>
</tr>
</tbody>
</table>
Table 264: Schedule Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the schedule.</td>
</tr>
<tr>
<td>Time Zone</td>
<td>Select the time zone for the schedule. If you select Floating, the time zone will be relative to whatever is accessing the item at any given time. For example, if a resource manager in Amsterdam sets a floating schedule for 8:00 A.M. to 5:00 P.M., a user in San Jose sees the schedule as 8:00 A.M. to 5:00 P.M. When a schedule is defined in a specific time zone, users in different time zones see the schedule with their own time zone applied.</td>
</tr>
<tr>
<td>Parent</td>
<td>Select a parent schedule to constrain the new schedule.</td>
</tr>
</tbody>
</table>
Define a schedule

Schedules are configured with two types of records.

- Schedule records specify a time zone and a type of schedule and use one or more schedule entries. Schedule records are saved in the Schedule [cmn_schedule] table.
- Schedule entry records specify the time periods that are included or excluded from a schedule. Schedule entries are saved in the Schedule Entry [cmn_schedule_span] table.

1. Navigate to System Scheduler > Schedules > Schedules.
2. Select a pre-existing schedule or click New to create a new one.
3. Complete the fields on the form (see table).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the schedule.</td>
</tr>
<tr>
<td>Time Zone</td>
<td>Select the time zone for the schedule. If you select Floating, the time zone will be relative to whatever is accessing the item at any given time.</td>
</tr>
<tr>
<td>Parent</td>
<td>Select a parent schedule to constrain the new schedule.</td>
</tr>
</tbody>
</table>
| Type        | Enter a label that describes the purpose of the schedule. You can also use one of these system terms to determine how to process certain schedules:  
  • excluded: excludes time periods from SLA counts.  
  • maintenance: specifies time periods where change management activities are allowed. A schedule containing maintenance schedule entries cannot also contain blackout schedule entries. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>black out</td>
<td>excludes time periods from change management schedules. A schedule containing blackout schedule entries cannot also contain maintenance schedule entries.</td>
</tr>
</tbody>
</table>

Description

[Optional] Describe the schedule.

**Note:** The Schedule form displays a warning message if there are no active entries defined for the current schedule. If your schedule is a child schedule that only contains exclusions, ignore the message because exclusions are non-active entries.

4. Right-click the header bar and click Save.

**Note:** If you create a schedule of type maintenance and save the record, a UI policy hides the Type field from the form. To view or change the value for the Type field, view the list of schedules rather than the schedule form and add the Type column if necessary. You can double click the cell for the value in the Type column and modify from the list view.

5. Configure one or more schedule entries.
## Schedule

**Name:** 8-5 weekdays  
**Time zone:** -- Floating --  
**Parent:**  
**Type:**  

### Description


#### Related Links

**Show Schedule**

**Schedule Entries (1)**

<table>
<thead>
<tr>
<th>Schedule Entries</th>
<th>Child Schedules</th>
<th>Referenced By</th>
</tr>
</thead>
</table>

### Schedule Entries

<table>
<thead>
<tr>
<th>Name</th>
<th>Repeats</th>
<th>Repeat every</th>
<th>Start date time</th>
<th>End date time</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday-Friday 8-5</td>
<td>Weekly on Weekdays</td>
<td>1</td>
<td>07-07-2008 08:00:00</td>
<td>07-07-2008 17:00:00</td>
<td></td>
</tr>
</tbody>
</table>

**Actions on selected rows...**
Schedule for the fifth instance of a day of the week

When selecting a date near the end of a month for a repeating monthly schedule, it is possible to select a date that computes to the fifth instance of that week day.

ServiceNow offers three options for handling months that do not have a matching fifth instance of the selected day.

- Last: ServiceNow selects the last instance of the week day in the month.
- Next: ServiceNow selects the first instance of the week day in the next month.
- Strict: ServiceNow skips any month without a matching fifth instance and selects only months that have a matching fifth instance.

The system property glide.schedules.fifth controls how a schedule entry that selects the fifth occurrence of a week day behaves in months containing only four occurrences of that day. This property is only valid when the glide.schedules.repeat_nth property is set to Day.

The following example illustrates computing what day of the month a schedule repeats on when the schedule starts on the fifth instance of a week day in the month.

1. Navigate to sys_properties.list.
2. Open the glide.schedules.fifth property.
3. Verify that the Value is set to last.
4. Navigate to System Scheduler > Schedules > Schedules, define a new schedule, and click Submit.
5. Open the new schedule and in the Schedule Entries related list, create a new entry with the following parameters:
   - When: November 29, 2012 at 10:00 to November 29, 2012 at 11:00
   - Repeats: Monthly
   - Monthly type: Day of the Week
   - Starting: November 29 (note that November 29 is the fifth Thursday in the month)
6. Click Submit.
7. Open the same schedule entry.
   Note that the form says "Every month on the fifth Thu."
Geneva    ServiceNow    ServiceNow Platform

The schedule for the first three months is computed as:

- November 29, 2012 (5th Thursday of the month)
- December 27, 2012 (Last Thursday of the month)
- January 31, 2013 (5th Thursday of the month)

8. If the Value on the glide.schedules.fifth property is set to next instead of last in step 3, the schedule for the first three months is computed as:

- November 29, 2012 (5th Thursday of the month)
- January 3, 2012 (1st Thursday of the next month since December 2012 does not have five Thursdays)
- January 31, 2013 (5th Thursday of the month)

9. If the Value on the glide.schedules.fifth property is set to strict instead of last in step 3, the schedule for the first three months is computed as:

- November 29, 2012 (5th Thursday of the month)
- No meeting (December 2012 skipped because it does not have five Thursdays)
- January 31, 2013 (5th Thursday of the month)

Repeat a monthly schedule

For monthly schedules (Repeat is set to Monthly) that start on a particular day of the month (Monthly type is set to Day of the month), you can specify the following options:

- How ServiceNow computes the starting day each month. See Day of the Week.
- How ServiceNow handles monthly schedules that start on the fifth instance of a day. See Fifth Instance of a Day of the Week.

Day of the Week:

ServiceNow offers two methods to compute what day of the week a monthly schedule repeats on:

- Day: This method computes the day of the week to repeat on by determining the order of the selected starting date within the month. For example, if the selected starting date appears on the first Monday in the month, the schedule repeats every first Monday of every month.
- Week: This legacy method computes the day of the week to repeat on by determining what week number the selected starting date appears in the month. For example, if the starting date is a Monday during the second week of the month, the schedule repeats the second Monday of every month.
The system property glide.schedules.repeat_nth determines what method your instance uses to compute what day a repeating monthly schedule occurs on. By default, instances use the more accurate Day method.

**Note:** Use the "Week" method to maintain backwards compatibility with customized schedule logic.

The following example illustrates computing what day of the week a monthly schedule repeats on.

1. Navigate to sys_properties.list.
2. Open the glide.schedules.repeat_nth property.
3. Verify that the Value is set to day.
4. Navigate to System Scheduler > Schedules > Schedules, define a new schedule, and click Submit.
5. Open the new schedule and in the Schedule Entries related list, create a new entry with the following parameters:
   - When: November 5, 2012 at 10:00 to November 5, 2012 at 11:00
   - Repeats: Monthly
   - Monthly type: Day of the Week
   - Starting: November 5 (note that November 5 is the first Monday in the month, but it is in the second week)
6. Click Submit.
7. Open the Schedule Entry.

   Note that the form says Every month on the first Mon.

   The first few dates this schedule will run are:
   - November 5, 2012 (1st Monday of the month)
   - December 3, 2012 (1st Monday of the month)
   - January 7, 2012 (1st Monday of the month)

8. If the Value on the glide.schedules.repeat_nth property is set to week instead of day in step 3, the first few dates this schedule will run are:
   - November 5, 2012 (Schedule starts on Monday in the 2nd week of the month)
   - December 10, 2012 (2nd Monday in the month)
   - January 14, 2012 (2nd Monday in the month)

**Time zones**

All times are stored in Coordinated Universal Time (UTC). They are displayed globally based on the system time zone, but are displayed to users in their local time zone according to user preferences.
Time zone representation

Time zones that have the Country/City format are primary time zone IDs. Other time zone IDs are links to the primary time zone. For example, US/Pacific is a link to the America/Los_Angeles time zone. Both America/Los_Angeles and US/Pacific represent Pacific Standard Time with the same zone offset and Daylight Savings Time (DST) schedule.

Other than the representation, there is no impact on date and time functionality.

In the absence of a default time zone for the user or the system, JVM reads default time zone information from the machine. Depending on how the machine is configured, it might return the Country/City or link, for example, US/Pacific or America/Los_Angeles. Administrators should configure their system with a default timezone "glide.sys.default.tz" to avoid system dependencies.

Daylight Saving Time

In general, if a time zone is specified based on location (for example, America/Los_Angeles), the system automatically adjusts for daylight saving time. If a time zone is specified based on the name of a time zone (for example, GMT), it does not adjust.

User preferences

Once the System Time Zone is defined, users can also select their own time zone from their user form, accessed through Self-Service > My Profile. The System default appears as System ([name of the default time zone]). For example, if the System time zone is America/Los_Angeles, the user sees System (America/Los_Angeles).

Time zones in email notifications

The date and time stamp of a notification uses the system time zone and not the time zone of any recipient. The property glide.email.append.timezone in System Properties > Email controls whether to append the time zone. If true, the system time zone of the instance is appended to any dates or date/times in outbound email messages (for example, 2010-07-02 04:01:14 PST).

Time zones in service level agreements

Service level agreements have different options for which time zone to use. To set a time-zone for SLAs, navigate to Service Level Management > SLA Properties and locate the following property:

![Use the following time zone for SLA:](image)

Figure 347: Time zone for service level agreements
Some special considerations:

- If **The caller's time zone** is selected, unpredictable behavior can occur if the caller does not have a time zone defined.
- If **The SLA definition's time zone** is selected, the time zone must be manually defined on the SLAs form.

**Time zone in scripting**

When scripting on the server, there are several GlideSystem Date and Time Functions used to get time values. For more information about specific methods and to learn the format in which each returns the requested time, see the GlideSystem API.

**Enhancements**

DST enhancement requires that reports and queries observe daylight saving time rules. The changes affect trend charts, line charts, and filters using the “trend on” operation.

**Time zone changer**

The time zone changer is active by default. Users can change their time zone in the system settings (the gear on the top right of the banner).

**Change the time zone choice list**

Wherever users have a choice of time zone, the choices are populated using the **Time Zone** choice list on the User [sys_user] table. Not all time zones appear by default.

To add or remove time zones from the list of time zones:

1. Navigate to **User Administration > Users** and open any user record, or click **New**.

   Notice the default time zone is **System (America/Los_Angeles)**.

2. Right-click **Time zone**, and then select **Personalize Choices**.

3. Highlight the desired time zone from the **Available** or **Selected** lists, and then **Add** or **Remove** the time zones as needed.
Change a time zone in a scheduled report

By default, scheduled reports use the time zone of the user who runs them (the user in the Run As field) to evaluate the query parameters. You are able to change this time zone.

For example, a report on 'incidents open today' resolves 'today' based on the user's time zone. To manually specify the time zone used to resolve the parameters of the report:

1. Configure the form to add the Run As tz field to the Scheduled Report form.
2. Select the appropriate time zone.

Change a time zone in a scheduled data import

By default, scheduled data imports are run using the time zone of the user who creates them. However, there is a way to manually specify the time zone for the import.

To change the time zone of the scheduled data import:

1. Configure the form to add the Run As tz field to the Scheduled Data Import form.
2. Select the appropriate time zone.
3. Click Update.

Set a system time zone

How to set a default time zone for calendars and users.

1. Navigate to System Properties > System.
2. Locate the property called System timezone, used as default for calendars and users.
   By default, the input field is blank. If no time zone is defined for this property, America/Los Angeles is used as the default.
3. Type a time zone in the field and click Save. This becomes the system time zone.
   The new time zone automatically cascades to all users who do not already have a specified time zone. If a user selects a different time zone or if the administrator selects a different time zone for them, the user is assigned the selected time zone and does not use the system time zone anymore.

© 2017 ServiceNow. All rights reserved. 1056
Time zones in SLAs

Service Level Agreements have a number of options of which time zone to use.

To set a time-zone for SLAs, navigate to Service Level Management > SLA Properties and locate the following property:

![Figure 348: Time Zone SLAs]

Some special considerations:

- If The caller’s time zone is selected, there will be unpredictable behavior if the caller does not have a time zone defined.
- If The SLA definition's time zone is selected, the time zone must be manually defined on the SLA’s form.

System log information

System logs track information that can be viewed by administrators.
The **System Logs** module provides access to a variety of system logs.

- Transactions: browser activities and background processes.
- Email: emails sent from within the platform.
- Events: events defined in the Event Registry.
- Imports: import activity within the platform.
- System Logs - warnings and errors triggered by the platform itself.

There are also log utilities provided:

- Log File Browser: browses the full log file.
- Log File Download: downloads the full log file.

**Audited tables**

If a field on a particular table is audited, all changes to that field are tracked.

For more information, see *Turning on Auditing (History) for a Table.*

This information is kept in two places:

- The *History sets* on page 2564 table.
- The *Audit table*.

**Configuration logging**

Certain configuration changes are tracked in the system logs.

- Action taken, including insert, update, and delete
- Category of change
- Comments recorded with the change
- Name of the change
- XML difference of the change
- Update set the change is associated to
- Date and time of the change
- User who made the change
- Table where the change was made
- Name of the object being changed
- Type of object being changed
- View the change was made in, for changes to forms or lists

**Customer updates table**

Every change that is made in the system is recorded on the Customer Updates [sys_update_xml] table chronologically.

To navigate to this table, enter `sys_update_xml.list` into the navigation filter.

The following information is stored about each update:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name that identifies the updated record.</td>
</tr>
<tr>
<td>Created</td>
<td>The date and time the customer update record was created.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Created By</td>
<td>The user who performed the change.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the update.</td>
</tr>
<tr>
<td>Updated</td>
<td>The date and time the customer update record was updated.</td>
</tr>
<tr>
<td>Updated By</td>
<td>The user who performed the update.</td>
</tr>
<tr>
<td>Updates</td>
<td>The number of times the record has been updated.</td>
</tr>
<tr>
<td>Target Name</td>
<td>The name of the element that was altered.</td>
</tr>
<tr>
<td>View</td>
<td>The view of the form that was altered if it was a form layout change.</td>
</tr>
<tr>
<td>Payload</td>
<td>The XML contents of the record after the change.</td>
</tr>
<tr>
<td>Remote Update Set</td>
<td>A reference to that update set if the change was performed by a remote update set.</td>
</tr>
<tr>
<td>Local Update Set</td>
<td>The update set the change is associated with.</td>
</tr>
</tbody>
</table>

### Email logs

The email log records all email notifications sent from all instances within the system.

This is a verbose and unfiltered view of email. For a more detailed view, see the *System Mailbox* application.

This log provides the following information for all notifications.

**Table 266: Email log**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox</td>
<td>The system mailbox to use for filtering the email notifications displayed.</td>
</tr>
<tr>
<td>State</td>
<td>The current state of the notification (Error, Ignored, Processed, or Ready).</td>
</tr>
<tr>
<td>Receive type</td>
<td>The type of inbound email notification (None, Forward, New, or Reply).</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>The status of the email notification. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• received: The server received this email.</td>
</tr>
<tr>
<td></td>
<td>• received - ignored: The server received this email, but it was ignored by the instance for inbound email action purposes. Typically,</td>
</tr>
<tr>
<td></td>
<td>these emails are either spam or auto-replies. See the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - failed: The server has attempted to send the email and failed. See the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - ignored: The server skipped sending this email. Typically, this is for an email which was generated but lacked a recipient email</td>
</tr>
<tr>
<td></td>
<td>address or is a duplicate email. See the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - ready: The email is ready to be sent, but has not been sent out by the mail server. Typically, an email remains in this state for</td>
</tr>
<tr>
<td></td>
<td>only a short time.</td>
</tr>
<tr>
<td></td>
<td>• sent: The email was sent by the instance without any errors or issues.</td>
</tr>
<tr>
<td>Target</td>
<td>A Document ID reference to the record if the email is generated by an insert, update, or delete of a particular record.</td>
</tr>
<tr>
<td>User</td>
<td>The name of the user, from the user record, of the instance from which the email notification was sent.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This is a string field.</td>
</tr>
<tr>
<td>Notification Type</td>
<td>The type of notification. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• None</td>
</tr>
<tr>
<td></td>
<td>• SMS</td>
</tr>
<tr>
<td></td>
<td>• SMTP</td>
</tr>
<tr>
<td>UID</td>
<td>The unique ID for the server.</td>
</tr>
<tr>
<td>Created</td>
<td>The date and time of the email activity for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Deleted</td>
<td>An indication of whether the email was deleted from an instance mailbox.</td>
</tr>
<tr>
<td>Weight</td>
<td>The weight of the email, which determines the sending priority relative to other notifications on the same table.</td>
</tr>
<tr>
<td>Importance</td>
<td>An indication that the email was sent with a changed level of importance, such as Urgent.</td>
</tr>
</tbody>
</table>
### Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Originating Event and Notification</td>
<td>An embedded list that stores the event and notification that initiated the email notification. For more information, see <em>Events</em> on page 2996.</td>
</tr>
<tr>
<td>Subject</td>
<td>A configured description of the action that generated the email notification. You create the subject text for notifications in <em>System Notification &gt; Email &gt; Notifications</em>.</td>
</tr>
<tr>
<td>Error String</td>
<td>The error string captured from the email server to determine why the email was not sent. This is logged only if the email is send-failed.</td>
</tr>
<tr>
<td>Recipients</td>
<td>The email address of the recipient of each notification.</td>
</tr>
<tr>
<td>Body</td>
<td>The body of the email, displayed in raw HTML markup. Use the related link <em>Preview HTML Body</em> to see the body text as rendered HTML.</td>
</tr>
<tr>
<td>Content type</td>
<td>The email content type.</td>
</tr>
<tr>
<td>Headers</td>
<td>Any headers embedded in the email.</td>
</tr>
</tbody>
</table>

Invalid email addresses that the instance strips out of outbound email messages are logged, starting with the Geneva release.

### Event logs

The event log records all system events that occur within the system.

This log provides the following information for all events that occur:

**Table 267: Event log**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the event for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the event. You configure events in <em>System Definition &gt; Business Rules</em>.</td>
</tr>
<tr>
<td>Parm1</td>
<td>Event-specific value that depends on the event and the recipient.</td>
</tr>
<tr>
<td>Parm2</td>
<td>Event-specific value that depends on the event and the recipient.</td>
</tr>
<tr>
<td>Table</td>
<td>Database table acted on for this event.</td>
</tr>
<tr>
<td>Processed</td>
<td>Date and time the event was processed This time reflects the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Processing time</td>
<td>Time taken to process this event, in milliseconds.</td>
</tr>
<tr>
<td>Queue</td>
<td>Processor queue name.</td>
</tr>
</tbody>
</table>
Import logs

The import log displays information in a verbose format about any data import activity within the platform. For a more detailed view of the import sets that produced a particular log, see Import Sets > Transform History.

This log provides the following information for all imports:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the import for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Level</td>
<td>Type of message displayed. For import files, the level is Information.</td>
</tr>
<tr>
<td>Message</td>
<td>System-generated message regarding the status of the import.</td>
</tr>
<tr>
<td>Source</td>
<td>Name of the external source of the import, such as an integration.</td>
</tr>
</tbody>
</table>

Logged information

Certain information is tracked in the system logs.

- Workflows
- Configuration
- Chats sessions
- Transactions for each view of each page in the system, including load times for network, server, and browser
- Inbound and outbound email
- Events triggered in the system
- Imports and integrations
- System warnings, errors, and script logs
- Upgrade information for any plugin activations, update sets, or system upgrades

Log history

The system uses table rotation and table extension to archive older logs.

By default, the system uses the following schedule to archive common logs:

<table>
<thead>
<tr>
<th>Table</th>
<th>Archive schedule</th>
<th>Rotations</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event [ecc_event]</td>
<td>Every day</td>
<td>7</td>
<td>Rotation</td>
</tr>
<tr>
<td>Queue [ecc_queue]</td>
<td>Every day</td>
<td>7</td>
<td>Rotation</td>
</tr>
<tr>
<td>Event [sysevent]</td>
<td>Every day</td>
<td>7</td>
<td>Rotation</td>
</tr>
<tr>
<td>Log [syslog]</td>
<td>Every week</td>
<td>8</td>
<td>Rotation</td>
</tr>
</tbody>
</table>
Log utilities

The instance provides the utilities log file browser and log file download.

Use System Logs > Log File Browser to view any system log entry. You can search for log files by using the following filters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start time</td>
<td>Start date and time of the range you want to search, for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Session ID</td>
<td>System-generated hexadecimal string that identifies the session that generated the log entry.</td>
</tr>
<tr>
<td>End time</td>
<td>End date and time of the range you want to search, for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Message</td>
<td>System-generated description of the occurrence.</td>
</tr>
<tr>
<td>Level</td>
<td>Type of message displayed. The levels are Debug, Error, Warning, and Information. A warning is an error that has been handled and recovered. An error is something that must be fixed.</td>
</tr>
<tr>
<td>Thread name</td>
<td>System-generated identifier of the thread that created the log file.</td>
</tr>
<tr>
<td>Max rows</td>
<td>Maximum number of records returned for a particular filter.</td>
</tr>
</tbody>
</table>

The instance creates compressed archives of system logs every two days and purges log archives after 21 days. You can download log file archives and view them with System Logs > Log File Download. Select a log archive from the list, and then click Download log under Related Links to open or save the archive.

**Note:** Log files are only available for the node you are currently logged into. To see the currently logged into node, navigate to System Diagnostics > Stats.

Outbound email notification recipients

For outbound notifications, the email system log provides reasons that recipients were included or excluded.

Each log entry corresponds to a reason for inclusion or exclusion. For example, all users who were excluded because they are inactive appear in a single log entry.
A series of system properties can be used to fine-tune the information to be logged. Two master switch properties, `glide.notification.recipient.include_logging` and `glide.notification.recipient.exclude_logging`, control all recipient inclusion and exclusion logging. Several other properties allow you to tailor the information reported in the logs to meet your needs. All of the properties are enabled by default.

![Outbound system logs](image)

**Figure 349: Outbound system logs**

**System diagnostics application**

The System Diagnostics application provides logs that relate to the platform.

These logs are available:

- **Upgrade History**: tracks every upgrade to an instance.
- **Slow Queries**: provides insight into how queries affect platform performance. See *Slow Query Logs*.

**System logs**

System logs display warnings and errors within the instance processes and records, and non-critical events such as memory usage on the server machine.

This list view displays the log entries for the current day only. To view other log files, use the log file browser.

This log provides the following information for all occurrences:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the logging activity for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Level</td>
<td>Type of message displayed. The levels are Debug, Error, Warning, and Information. A warning is an error that has been handled and recovered. An error is something that must be fixed.</td>
</tr>
<tr>
<td>Message</td>
<td>System-generated message regarding the nature of the occurrence.</td>
</tr>
</tbody>
</table>
### Email logs

The Emails module links to all of the mail received or sent by the platform.

For more information, see *System Mailboxes*.

### Transaction logs

The transaction log records all browser activity for an instance.

This log provides the following information for all activities.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Name of the process or area affected by the occurrence. For example, the source of the occurrence might be EMAIL or Memory.</td>
</tr>
</tbody>
</table>

#### Table 272: Transaction log

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the browser action for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Created by</td>
<td>The user who created this activity.</td>
</tr>
<tr>
<td>Response time</td>
<td>Round trip response time for the browser request, in milliseconds.</td>
</tr>
<tr>
<td>Network time</td>
<td>Latency time of the network response after the browser request is made, in milliseconds.</td>
</tr>
<tr>
<td>Output length</td>
<td>Size of the output string sent by the instance to the browser, in bytes.</td>
</tr>
<tr>
<td>SQL count</td>
<td>Number of SQL server commands executed for this activity.</td>
</tr>
<tr>
<td>Business rule count</td>
<td>Number of business rules executed for this activity.</td>
</tr>
<tr>
<td>Business rule time</td>
<td>Elapsed time for the execution of the business rules for this activity.</td>
</tr>
<tr>
<td>URL</td>
<td>The application or module connected to by the client browser.</td>
</tr>
<tr>
<td>System ID</td>
<td>System generated identifier of the client instance making the request. This ID is used for cluster environments in which several instances (nodes) communicate with the database.</td>
</tr>
<tr>
<td>IP address</td>
<td>IP address of the client making the request.</td>
</tr>
<tr>
<td>GZipped</td>
<td>Indication of whether a compressed Web page was requested by the browser.</td>
</tr>
<tr>
<td>Protocol</td>
<td>The HTTP protocol used by the browser for this instance.</td>
</tr>
</tbody>
</table>
The **System Scheduler > Slow Job Log** module provides a transaction log filtered to show only slow transactions.

**Workflow logging**

Certain workflow information is tracked in the system logs.

- Each activity executed, including:
  - Date and time started
  - Date and time ended
  - State, for example, Finished, Cancelled, Timed Out, Error
  - Result
  - Fault description, if there was an error

- Transition history, including:
  - Time of transition
  - Activity transitioned from
  - Activity transitioned to
  - Which transition was triggered

- Workflow log, including any log statements added to the workflow

**Search administration**

The system uses the Zing search engine to manage search functionality.

### Explore
- **ServiceNow Platform release notes**
- **Upgrade to Geneva**
- **Features of search administration** on page 1066
- **Features of Zing text indexing and search engine** on page 1068

### Set up
- **Set global text search properties** on page 1074
- **Zing can include attachments in search results** on page 1102

### Administer
- **Zing indexes words** on page 1094
- **Global text search finds records from multiple tables** on page 1069
- **List search finds records from the current table** on page 1081

### Use
- **Boolean operators allow conditional search results** on page 1083
- **Quotation marks allow exact phrase searches** on page 1085
- **Wildcard characters allow searching for patterns and variations** on page 1086

### Develop
- **Developer training**
- **Developer documentation**
- **Installed with Zing** on page 1107

### Troubleshoot and get help
- **Ask or answer questions in the Platform community**
- **Search the HI Knowledge Base for known error articles**
- **Contact ServiceNow Support**

**Features of search administration**

Enable and configure search administration features.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Top tasks</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Zing text indexing and search engine</strong> on page 1067</td>
<td>Index and search record data by table.</td>
<td>• <a href="#">Enable text indexing for a table</a> on page 1097</td>
<td>• Active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Zing can include attachments in search results on page 1102</td>
<td></td>
</tr>
<tr>
<td><strong>Google custom search integration</strong> on page 1109</td>
<td>Replace Zing text indexing and search engine with Google Site Search functionality. Requires configuration before use</td>
<td>• <a href="#">Block a robot from your site</a> on page 1113</td>
<td><a href="#">Available by request from ServiceNow personnel</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Configure a search property</a> on page 1109</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Define a custom search page</a> on page 1110</td>
<td></td>
</tr>
</tbody>
</table>

### Zing text indexing and search engine

Index and search record data by table.

Zing text search is:

- an index split into multiple database shards for parallel querying
- created entirely with ServiceNow code
- implemented purely within the relational model
- yields fast results with search engine logic
- provides search results with an emphasis on relevancy scoring
  - frequency points
  - proximity points
  - field-level scoring (such as title, metadata, and short description)
- provides search suggestions
  - Type-ahead search suggestions
  - **Did you mean:** global search suggestions

Administrators and users with the ts_admin role can configure:

- match relevance
- field weights for each table
- global stop words
- stop words for each index
- synonym dictionaries for each table
- tables with attachment indexing

By default, the system uses the Zing text indexing and search engine to search record data.
Features of Zing text indexing and search engine

Enable and configure Zing text indexing and search engine features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Top tasks</th>
<th>State</th>
</tr>
</thead>
</table>
| Zing computes document scores using three components on page 1092 | Compute document scores based on the frequency, sequence, and weight of search terms in the document. | • Set the relative weight of a field on page 1093  
  • Enable text indexing for a table on page 1097 | Active                  |
| Global text search finds records from multiple tables on page 1069 | Search multiple record types from a single search field.                   | • Add a search group on page 1072  
  • Enable text indexing for a table on page 1097 | Active                  |
| List search finds records from the current table on page 1081  | Search records from a table list view.                                      | • Enable text indexing for a table on page 1097  
  • Regenerate a text index for a table on page 1098 | Active                  |
| Zing can include attachments in search results on page 1102    | Expand search results to include attachments from indexed tables.           | • Enable attachment indexing on a table on page 1104  
  • Enable text indexing for a table on page 1097 | Active                  |
| Zing removes stop words from queries on page 1104             | Remove common words from search queries that do not produce meaningful results. | • Configure a global stop word on page 1105  
  • Configure a table-specific stop word on page 1105 | Active                  |
| Zing matches derived words with stemming on page 1107         | Convert any multiple-character search keyword to its stem form to find derived versions of the word. | • Set localization properties on page 984  
  • Activate a language on page 977 | Active                  |

Available search options

Zing offers users several search interfaces and search operators to conduct text searches.
Table 273: Available search interfaces

<table>
<thead>
<tr>
<th>Search interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global text search</td>
<td>Find records in multiple tables from a single search field.</td>
</tr>
<tr>
<td>Lists</td>
<td>Find records in a list; search in a specific field (Go to), all fields (Search), or in a specific column.</td>
</tr>
<tr>
<td>Knowledge Base</td>
<td>Find knowledge articles.</td>
</tr>
<tr>
<td>Navigation filter</td>
<td>Filter the items in the application navigator.</td>
</tr>
<tr>
<td>Live feed</td>
<td>Filter, search, or sort messages in live feed.</td>
</tr>
<tr>
<td>UI pages on page 3869</td>
<td>Create a custom UI page to search for records in a table.</td>
</tr>
</tbody>
</table>

Table 274: Available search operators

<table>
<thead>
<tr>
<th>Search operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean operators</td>
<td>Include or exclude search terms from global text searches.</td>
</tr>
<tr>
<td>Quotation marks</td>
<td>Filter search results to only include matches to an exact phrase consisting of one or more words.</td>
</tr>
<tr>
<td>Wildcard characters</td>
<td>Expand search results to match any non-space character.</td>
</tr>
<tr>
<td></td>
<td>The system supports both single-character and multiple-character wildcards.</td>
</tr>
</tbody>
</table>

Global text search finds records from multiple tables
Search multiple record types from a single search field.

To perform a global text search, perform one of the following actions based on the UI version you are using.

<table>
<thead>
<tr>
<th>UI version</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI16</td>
<td>Click the search icon in the banner on the upper right of the UI16 banner frame on page 18, and then enter the search term.</td>
</tr>
<tr>
<td>UI15 or UI11</td>
<td>Enter the search term in the search box in the upper right of the UI15 banner frame on page 21 or UI11 banner frame on page 26.</td>
</tr>
</tbody>
</table>

All UI versions also have a global text search keyboard shortcut. The UI global_text_search UI Macro provides the global text search field.

Search groups filter search results by table
The system displays search results by search group, which are collections of related tables that users can enable or disable to filter search results.

Within each search group, the system divides search results by table. A user must have read access to a table to see search results for it.
Default search groups

Table 275: Default system search groups

<table>
<thead>
<tr>
<th>Search group</th>
<th>Example tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks</td>
<td>Incident, Change Request, Problem</td>
</tr>
<tr>
<td>Live Feed</td>
<td>Live Feed Message</td>
</tr>
<tr>
<td>Policy</td>
<td>Business Rule, UI Policy, Client Script</td>
</tr>
<tr>
<td>People &amp; Places</td>
<td>User, Group, Location, Company</td>
</tr>
<tr>
<td>Knowledge &amp; Catalog</td>
<td>Knowledge Base, Service Catalog</td>
</tr>
</tbody>
</table>

Displaying search results by group

The system displays knowledge and service catalog search results within their respective search results pages. The system displays other results in list format.

For exact match searches, the system only returns records from the Task (task) and Knowledge (kb_knowledge) tables.

**Note:** The system filters global search results to display only knowledge articles that are in the same language as the preferred language setting.
Options for administrators

Administrators can customize these search groups settings for all users:

- Specifying which tables are part of the group.
- Specifying whether the search group is available for use.
- Specifying whether searches include results from this search group by default.
- Creating new search groups.

Enabling a search groups allows users to see records from the search group's tables. Disabling a search group hides records from the search group's tables.

Options for users

Users can specify these individual search group preferences:

- Specify which group and table search results are collapsed and expanded by default.
• Specify which groups and tables the system includes in search results.

You can collapse or expand each search group and search table as you like, and those settings can be used for subsequent searches if you choose. For example, if you are rarely interested in Policy or Core Items search results, but still want to search them each time, you can collapse those groups.

If you do not want to search some groups or tables, you can deselect a search group or table on a per-user basis. Your preference is saved for subsequent searches. To deselect a search group, clear its check box on the search results page.

**Figure 351: Deselecting search groups**

Click a search group's link, such as the blue **Tasks** link in the image above, to display a dialog box where you can deselect specific tables in each group.

**Figure 352: Deselecting search tables**

Add a search group

You can add a search group to filter global search results.

Role required: text_search_admin

1. Navigate to **System Definition > Search Groups**.
2. Click **New**.
3. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name of the search group.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
Roles | Select one or more user roles if you want to restrict access to the search group to those roles.
User group | Select a user group if you want to restrict access to the search group to users in that user group.
Description | Enter text to display as a tooltip for the search group.
Active | Ensure the check box is selected to make the search group available.
Searched | Clear the check box if you want to exclude results from this search group by default. A user can manually choose to show results from this search group on the search results page.
Order | Enter a number to determine the order in which this search group is listed relative to other search groups.

4. Right-click the form header and select **Save**.
5. In the **Text Search Tables** related list, click **New**.
6. Complete the form.

**Table 277: Text search table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select the table to search.</td>
</tr>
<tr>
<td>Active</td>
<td>Ensure the check box is selected to search the table.</td>
</tr>
<tr>
<td>Searched</td>
<td>Clear the check box if you want to exclude results from this table by default. A user can manually choose to show results from this table on the search results page.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to determine the order in which this table is listed relative to other tables in the search group.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Select one or more conditions to determine which records on the selected <strong>Table</strong> are searched.</td>
</tr>
<tr>
<td>Optional label</td>
<td>Enter the label to display for the table. For example for a search group intended for your Sales team, display <strong>Contacts</strong> instead of <strong>Users</strong> for the <strong>User [sys_user]</strong> table.</td>
</tr>
</tbody>
</table>

7. Click **Submit**.
8. Add any other text search tables to include in the search group.

Set global text search properties
Administrators can control how global text search behaves with system properties.

Role required: admin
1. Navigate to System Properties > Global Text Search.
2. Set these properties.

Table 278: UI properties for global text search

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Global Text Search glide.ui.text_search.enable</td>
<td>The default value is true. Clear this option to disable the global text search capability.</td>
</tr>
<tr>
<td>List of roles (comma-separated) that can use the Global Text Search capability. glide.ui.can_search</td>
<td>The default value is itil,text_search_admin,admin. Add a role to enable it to use search.</td>
</tr>
<tr>
<td>List of tables (comma-separated) that will not appear as options when setting up Global Text Search tables. glide.ui.no_text_search</td>
<td>The default value is sys_audit,sys_event,ecc_queue,ecc_event,syslog,syslog_transaction,sys_journal_field,sys_audit_relation,ecc_agent_log,ecc_mi_result,sysrule_escalate_history,sys_user_token,sys_time_dimension,sys_attachment_doc. Tables that do not appear as options when defining search groups. Examples are system or maintenance tables.</td>
</tr>
<tr>
<td>Number of Global Text Search matches returned per table glide.ui.text_search.rowcount</td>
<td>The default value is 10. Users can click through to see all results for a specific table.</td>
</tr>
<tr>
<td>Global Text Search background color for Knowledge Base results css.textsearch.kb.background.color</td>
<td>The default value is #FFFFDD.</td>
</tr>
<tr>
<td>Global Text Search background color for Catalog results css.textsearch.catalog.background.color</td>
<td>The default value is #F0F7F9.</td>
</tr>
<tr>
<td>Form view to use for Global Text Search exact match - blank means Default view glide.ui.text_search.match_view</td>
<td>The default value is blank.</td>
</tr>
<tr>
<td>List and form view to use for Global Text Search results. Blank means default view. glide.ui.text_search.view</td>
<td>The default value is the text_search view. If you do not have a text_search view, you can create it.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>glide.ir.query_method</td>
<td>Query method for global text search. Run a query for each table, for each base class (parent table), against a single index regardless of table (requires text index regeneration), or let GlideRecord handle the query (simple). With the Zing text search engine, the preferred value is &quot;simple&quot;. The default value is <strong>Simple query</strong>.</td>
</tr>
<tr>
<td>glide.lucene.base_result_max</td>
<td>The maximum number of results that will be returned during a text index search. This limits the number of records that will be extracted from the text index and displayed. The default value is 100. This property is not supported by the Simple query method and is ignored.</td>
</tr>
<tr>
<td>glide.lucene.base_hits_max</td>
<td>The maximum number of entries that will be retrieved from the text index. Only the table that is involved will be returned from the text index for all entries that are between the value specified for maximum results and this value. This is done so that it can be indicated that more results exists for any given table that are going to be displayed. The default value is 500. This property is not supported by the Simple query method and is ignored.</td>
</tr>
<tr>
<td>glide.ts.global_search.parallelism</td>
<td>Number of simultaneous processes (1 to 16) used when searching though multiple groups in a global search. The default value is 4. Each search group uses one thread to render results. Set this value to yield optimal results for your search group configuration. For example, if you have five search groups and four threads, the first four groups run in parallel and the fifth group starts when one of the first four groups finishes. This setup may work well if one of the groups is much larger than another. Similarly, if you have five search groups, setting this value higher than five yields no benefits.</td>
</tr>
</tbody>
</table>

3. Click **Save**.

   The business rule **Text Search Property Change Rationally** validates the new values and aborts the update if they are outside its acceptable ranges.

4. **Open the System Property table** and search for the property **glide.ts.max_wildcard_expansion**.

   **Table 279: System Property for wildcard searches**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.ts.max_wildcard_expansion</td>
</tr>
<tr>
<td>Description</td>
<td>Maximum number of matches for a wildcard term that are allowed as unambiguous.</td>
</tr>
</tbody>
</table>
5. Edit the Value.
   If a wildcard search matches more words than this value, the system displays a message asking the user to refine the search.

6. Click Update.

7. To set the text search stemming language, install an internationalization (i18n) plugin.

8. Navigate to **System Properties > System Localization**.

9. Set this property.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Value</td>
<td>500</td>
</tr>
</tbody>
</table>

### Table 280: Text stemming property

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Search stemming language glide.ts.stemming_language</td>
<td>The default value is <strong>English</strong>. The system supports stemming in English and German.</td>
</tr>
</tbody>
</table>

Enable the simple query method

Configure the simple query method to allow GlideRecord to handle queries and is supported by Zing Text Search.

1. Navigate to **System Properties > Global Text Search**.
2. Locate **Query method for global text search** (the glide.ir.query_method property).
3. Select **Simple query**.
4. Click **Save**.

   The following properties are not supported:
   - Maximum number of results that are displayed per table (glide.lucene.base_result_max)
   - Maximum number of results counted per base class (glide.lucene.base_hits_max)

Configure parallel processing of search groups

To improve performance, only activate search groups and tables that are necessary to meet business needs.

For example, if you do not need Change Task results, deactivate that table in the Tasks search group. If only one group of users needs Change Task results, set up a separate search group that includes Change Tasks. Other users search using a group that does not contain Change Tasks.

Global text search can render results in parallel to improve performance. To configure the number of parallel processes:

1. Navigate to **System Properties > Global Text Search**.
2. Locate the property called **Number of simultaneous processes (1 to 16) used when searching though multiple groups in a global search**.
3. Enter the number of processes to run in parallel.

   Each search group uses one thread to render results. Set this value to yield optimal results for your search group configuration. For example, if you have five search groups and four threads, the first four groups run in parallel and the fifth group starts when one of the first four groups finishes. This
setup may work well if one of the groups is much larger than another. Similarly, if you have five search groups, setting this value higher than five yields no benefits.

4. Click 

Disable the junk filter

By default, Zing does not index or search for 2-digit numbers and single character words (unless they are Chinese or Japanese characters). You can turn off this filter for a table.

For example, to have the ability to search for error 92 in the knowledge base, turn off junk filtering for kb_knowledge.

1. Navigate to System Definition > Dictionary.
2. Open the dictionary entry for the table.
3. In the Attributes field, enter text_index_filter_junk=false.
4. Right-click the header and select Save.
5. Click the Generate Text Index related link and click OK.

Note:

• To have the desired effect, you may also need to reconfigure the automatic stop words. For example, if a specific 2-digit number appears in all of your problem numbers (92-0001, 92-0002, and so on), the automatic stop word threshold is quickly exceeded and the word is no longer found in searches. To find that 2-digit number in the problem table, you must disable both junk filtering and automatic stop words.
• Disabling the junk filter results in a larger table index. For optimal performance, do not apply this attribute unless it is required.

Debug Zing

Debugging messages create a log of search processes as they occur. These messages may help administrators configure optimal performance settings.

• To view debugging messages, navigate to System Diagnostics > Session Debug > Debug Text Search. The search log messages appear at the bottom of the results page for the remainder of your session.

Figure 353: Debugging Messages

• To debug text indexing problems, you may want to view text_index events in the system log.
In rare cases, you may need to reset the text search caches after regenerating a text index. If text search fails for known words, diagnose and fix this problem:

1. Enable text search debugging messages.
2. In any table with search problems, search for the problem text.
3. In the debug output, note the numbers used for the Stemmed terms.
4. Enter `ts_word.list` in the navigation filter.
5. Locate the stemmed terms and compare the numbers in the list to the debug output (Step 3). If the numbers are different, the UI node's cache is stale.
7. Click the Reset Text Search Caches related link and click OK.

Global search user preferences

The global text search results page allows users to set several user preferences for global text search. The following search preferences are available from the Search tips and preferences link.

![Search tips and preferences](image)

Figure 354: Search preferences
<table>
<thead>
<tr>
<th>Label</th>
<th>Preference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use remembered expand / collapse preferences</td>
<td>ts.remember.expanded</td>
<td>Specifies whether you want the collapse state of any search groups/tables remembered for your next search. If selected and you collapse the search results for the People &amp; Places search group for example, that group will be collapsed on your next search. If cleared, all groups and tables are expanded for every search.</td>
</tr>
<tr>
<td>Show list of tables with no search matches</td>
<td>ts.show_negative_result_info</td>
<td>Specifies whether or not you want to see a summary of tables that had no search matches. For example: No matches for Requests, Catalog Tasks, Tickets.</td>
</tr>
<tr>
<td>Show selectable search groups</td>
<td>ts.show_search_groups</td>
<td>Shows or hides the search group check box row.</td>
</tr>
<tr>
<td>Show groups with no search matches</td>
<td>ts.show_empty_groups</td>
<td>Shows or hides a search group if it returns no matches. If selected, the search results display a placeholder row stating there were no matches for that group.</td>
</tr>
<tr>
<td>Return task record if searching for exact number</td>
<td>ts.match</td>
<td>Shows or hides the full search results page if the search term matches a task record number. When selected, an exact match to a record number causes the system to hide the full search results and instead display the record with an exact match message. Exact search matches ignore most additional search conditions normally applied to the record such as the currently selected search groups or the knowledge article published state. For example, an exact search match for a knowledge article returns the article in any state as long as the current user has read access to it.</td>
</tr>
</tbody>
</table>

**Note:** The system filters exact search results to display only knowledge records that are in the same language as the preferred language setting.
Global text search suggestions

Global text search offers two types of search suggestions.

The knowledge base and global text searches provide suggestions as you type. Type-ahead suggestions appear under the search box. Suggestions are based on similar searches that begin with the same characters.

![Type-ahead search suggestions](image)

**Figure 357: Type-ahead search suggestions**

The knowledge base and global text searches also provide suggestions for alternate search terms. Suggestions appear if your original search does not return any results and an alternate spelling or similar recent search does. For example, if you misspell a search term (such as eail), the correct spelling (email) may appear as an alternate search suggestion. Suggestions appear beneath the search box on the results page.

![Alternate search suggestions](image)

**Figure 358: Alternate search suggestions**

Update a type ahead suggestion

The knowledge base and global text searches provide suggestions as you type. These type-ahead suggestions are compiled on a nightly basis by a scheduled job.

Use the following procedure if you need to refresh this list sooner.

1. Navigate to **System Scheduler > Scheduled Jobs**.
2. Open **TS Search Stats**.
3. Run the scheduled job.

For more about how suggestions are generated and maintained, see the blog post *Global Text Search Suggestions* by a ServiceNow Technical Support Engineer in the ServiceNow Community.

Configure a "Did You Mean?" suggestion

The Knowledge Base and global text searches can provide "Did you mean?" suggestions. Suggestions appear if a search does not return any results and an alternate spelling or similar recent search does. This feature is disabled by default.

1. Navigate to **System Properties > Text Search**.
2. Select any of the following check boxes:
   - **Suggest alternate search spellings for knowledge or global search** (the glide.ts.dym.enable_spell_correct property).
• **Suggest related searches for knowledge or global search** (the glide.ts.dym.enable_chain_suggest property); uses search chains, which generate suggestions by tracking occurrences of similar searches, in order, over time.

3. Click **Save**.

*List search finds records from the current table*
Search records from a table list view.

Indexed tables display the for text option in the list title bar, which searches all records for matching field values.
### INCIDENT CONDITIONS

All of these conditions must be met

Keywords are

or

New Criteria

### RELATED LIST CONDITIONS

<table>
<thead>
<tr>
<th>Number</th>
<th>Created</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC0000051</td>
<td>2016-11-15 12:48:32</td>
<td>Manager can't access SAP Controlling application</td>
</tr>
<tr>
<td>INC0000052</td>
<td>2016-11-15 12:48:40</td>
<td>SAP Financial Accounting application appears to be down</td>
</tr>
<tr>
<td>INC0000055</td>
<td>2016-11-15 20:47:23</td>
<td>SAP Sales app is not accessible</td>
</tr>
<tr>
<td>INC0000053</td>
<td>2016-11-15 12:48:46</td>
<td>The SAP HR application is not accessible</td>
</tr>
<tr>
<td>INC0000054</td>
<td>2015-11-02</td>
<td>SAP Materials Management is slow or</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
The list search field accepts *Boolean operators* (AND, OR, and NOT) in search queries. When a user adds a Boolean operator to a search query, the system only returns records that match all search conditions of the query.

The system also converts any search query into an equivalent keyword condition in the list breadcrumbs and filter. For example, searching for the text "Oracle OR SAP" produces the condition `[Keywords] [are] [Oracle OR SAP]`. The standard list controls can modify or remove these breadcrumbs and conditions.

*Boolean operators allow conditional search results*
Include or exclude search terms from global text searches.

All global text search interfaces support boolean operators.

- Global text search field
- Knowledge Base search
- List search for text

For Knowledge Base searches, if the AND search returns poor results, the search is automatically re-run with the OR operator. Administrators can configure the knowledge base search to always use OR by modifying the `glide.knowman.search.operator` property.

**Note:** Boolean operators are case-sensitive. For example, OR is an operator while or is a search term that may be a stop word.
Table 282: Boolean operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR or vertical bar symbol (</td>
<td>Finds a match if any of the terms exist in a document (a union using sets). For example, to find documents that contain either &quot;email password&quot; or just &quot;email&quot;, search for &quot;email password&quot; OR email or &quot;email password&quot;</td>
</tr>
<tr>
<td></td>
<td>When you use filter lists on keywords, OR and</td>
</tr>
<tr>
<td></td>
<td>produce the same results. However, using the .or. operator, which the system uses when you create multiple conditions with the condition builder, can produce different results. For example, these two filters produce the same result by searching for records that contain either SAML or SSO:</td>
</tr>
</tbody>
</table>

![Figure 359: Searching with the OR operator](image1)

![Figure 360: Searching with the pipe operator](image2)

But this filter may produce a different result as it searches for records containing SAML and then searches again for records containing SSO:
<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
</table>
| **AND** | Finds a match if all terms exist in a document (an intersection of sets). For example, to find documents that contain "CPU load" and "10 minutes", search for "CPU load" AND "10 minutes".  
By default, Zing applies an AND operator between all search terms regardless of whether the user entered them or not. For example, the system converts a search for the text "email server down" to [Keywords] [are] [email] AND [Keywords] [are] [server] AND [Keywords [are] [down]]. |
| **NOT, minus (-), or exclamation point (!)** | Excludes documents that contain the term after the NOT operator (a difference of sets). For example, to find documents that contain "CPU load" but not "10 minutes", search for "CPU load" NOT "10 minutes", "CPU load" -"10 minutes", or "CPU load" !"10 minutes".  
The NOT operator cannot be used with just one term. For example, the following search returns no results: NOT "10 minutes".  
NOT must be a stand-alone word. For example, atom NOT ion excludes the term ion. Whereas NOTION searches for the term notion.  
Minus and exclamation point must immediately precede the excluded term. For example, email !Joe excludes the term Joe but email ! Joe includes the term Joe. |

**Quotation marks allow exact phrase searches**
Filter search results to only include matches to an exact phrase consisting of one or more words.

Zing only matches documents that contain the exact words in the exact order you specify. An exact phrase search ignores:

- stop words
- punctuation marks
- wildcard characters

**Note:** Wildcard characters are ignored as punctuation.

For example, searching for the phrase "email password" returns the same search results as these phrases:

- "email the password" because the stop word "the" is ignored.
- "email password?" because the punctuation "?" is ignored.
- "email password*" because the wildcard "*" is ignored as punctuation.
**Wildcard characters allow searching for patterns and variations**

Expand search results to match any non-space character. The system supports both single-character and multiple-character wildcards.

The following searches support wildcard characters.

- Global text search
- Knowledge Base
- Lists (text searches of all fields)
- Reference fields

**Note:** Wildcard text searches are only effective with multiple character words. A wildcard within a sequence of pictogram characters returns too many options for an efficient search. Wildcards are best used with letter-based words.

To perform a single-character wildcard search, use the percent sign (%) character. This wildcard finds words that contain any one character in place the percent-sign-character. For example, to find words such as text or test, search for: te%t

To perform a multiple-character wildcard search, use the asterisk (*) character. This wildcard finds words that contain zero or more characters in place of the asterisk-character. For example, to find words such as planned or placed, search for: p*l*d

You can use wildcard characters anywhere in a search string. If a wildcard search returns too many hits, the system displays a message asking you to refine the search. Administrators can set a global text search property to change the limit for wildcard search results.

**Zing generates search results in four phases**

Generate search results by dividing documents among shards, searching shard documents, scoring shard documents, and merging scores into an index.

Zing generates search results in four phases:

1. Divide searchable documents among index shards.
2. Search and filter shard documents.
3. Score shard documents.
4. Merge shard document scores into an index.

**Phase 1: Divide searchable documents among index shards**

When a search is executed, the system equally divides all searchable documents among 10 index shards. Each shard has a unique list of documents.

To maximize search efficiency, the system creates 10 query threads to simultaneously search each index shard.
Phase 2: Search and filter shard documents

Each query thread searches through its list of shard documents to identify documents that match the search criteria. For example, if you search for "wifi network," the system returns all documents containing both "wifi" and "network" in no particular order.

Note: A multiple word search is equivalent to multiple single word searches connected by an AND operator. For example, `[contains][wifi][AND][contains][network]`.

The query thread assigns each matching document a numeric value (a document ID) to uniquely identify it. All other documents are ignored.

Phase 3: Score shard documents

The query thread scores each matching shard document.
Figure 363: Query threads search, filter, and score shard documents
## Phase 4: Merge and sort shard document scores

The system merges the document scores into a single index and sorts the documents from highest to lowest document score. The documents with the highest document score are most relevant to the search query.

### Figure 364: Merge and sort shard document scores

<table>
<thead>
<tr>
<th>Doc</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
<td>11</td>
</tr>
<tr>
<td>91</td>
<td>6</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>12</td>
</tr>
<tr>
<td>81</td>
<td>11</td>
</tr>
<tr>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>91</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>
Zing filters search results with access controls

Filter search results to only display records the user can access.

For example, suppose you index the System Properties \([\text{sys}_\text{properties}]\) table. When the ITIL User searches for a term in the system properties table, Zing returns no search results because the ITIL user does not meet the ACL rule requirements.

Figure 365: ITIL User Search

When a system administrator searches for the same property, Zing returns search results from the System Properties tables because the administrator meets the ACL rule requirements.
### System Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Type</th>
<th>Description</th>
<th>Updated</th>
<th>Updated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.smtp.default_retry</td>
<td>true</td>
<td>true</td>
<td>Resend email when unknown SMTP error code</td>
<td>2013-01-27</td>
<td>admin</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>false</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.email.user</td>
<td></td>
<td>string</td>
<td>User email (eg. <a href="mailto:helpdesk@company.com">helpdesk@company.com</a>)</td>
<td>2013-01-04</td>
<td>guest</td>
</tr>
<tr>
<td>glide.smtp.auth</td>
<td>false</td>
<td>true</td>
<td>Authenticate with the SMTP server using</td>
<td>2010-06-23</td>
<td>dico</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>false</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.smtp.active</td>
<td>true</td>
<td>true</td>
<td>Enable email sending (SMTP).</td>
<td>2009-08-26</td>
<td>glide.maint</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>false</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.smtp.fail_message</td>
<td>500,501,502</td>
<td>string</td>
<td>Do not resend email if these SMTP error codes a...</td>
<td>2012-01-27</td>
<td>admin</td>
</tr>
<tr>
<td>glide.smtp.delete_retry</td>
<td>421,450,451</td>
<td>string</td>
<td>Resend email if these SMTP error codes a...</td>
<td>2012-01-27</td>
<td>admin</td>
</tr>
<tr>
<td>glide.smtp.defor_retry</td>
<td>452</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.smtp.server</td>
<td>relay</td>
<td>string</td>
<td>Outgoing (SMTP) mail server.</td>
<td>2013-03-06</td>
<td>cory.seering@enc</td>
</tr>
<tr>
<td>glide.smtp.default_suffix</td>
<td></td>
<td>string</td>
<td>Default gateway address for any out bound</td>
<td>2007-03-28</td>
<td>glide.maint</td>
</tr>
<tr>
<td>glide.pop3.user</td>
<td></td>
<td>string</td>
<td>incoming (POP) mail account name, if different...</td>
<td>2013-01-04</td>
<td>guest</td>
</tr>
<tr>
<td>glide.pop3.server</td>
<td></td>
<td>string</td>
<td></td>
<td>2013-01-04</td>
<td>guest</td>
</tr>
</tbody>
</table>

### People & Places

No matches for Users, Groups, Locations, Companies.

### Knowledge & Catalog

No matches for Knowledge Bases, Catalog Items.
Zing computes document scores using three components

The Zing search engine computes document scores based on the frequency, sequence, and weight of search terms in the document.

The components of a document score are:

- **Frequency**: how often the search terms appear in the document.
- **Sequence**: how often the search terms appear in the same order as the search query.
- **Weight**: how heavily weighted the source field is in which the search terms appear.

**Figure 367: Sample document score computation**

**Frequency points**

Zing awards one point whenever a search term appears anywhere in the document. For example, when searching for "distributed database server", a document that contains "distributed" three times, "database" five times, and "server" 17 times would have 25 frequency points.
Sequence points

Zing awards a document more points when it contains the search terms in the same order in which they were typed. The more search terms in sequence there are, the exponentially higher the score becomes.

Following the example above, each time the string "database server" appears in a document, it is awarded 100 (10^2) sequence points. Likewise, each time the string "distributed database server" appears in a document, it is awarded 1000 (10^3) sequence points.

Zing awards sequence points as 10^x, where x is the number of search terms that appear in sequence.

Weight points

Zing awards a document more points if the field in which it appears is weighted. Zing applies a multiplier to frequency points based on the value of the ts_weigh attribute for the field.

By default, each field has a weight of 1. A field with a weight of 50 (ts_weight=50) would add 50 points each time a search term appeared in the field. The maximum possible weight value is 255.

In the out of box system, the following fields have elevated weight scores:

- kb_knowledge.number = 50
- kb_knowledge.short_description = 10
- kb_knowledge.meta = 10
- task.number = 50
- task.short_description = 10

Set the relative weight of a field

To improve search results, the search engine assigns to each potential match a numeric score that represents its relevancy to the query.

Role required: admin

Administrators can control the relative importance of a match for each field in a table with the ts_weight attribute. The default weight of a field is 1.

1. Open the system dictionary entry for the field.
2. In the Attributes field, enter ts_weight=<value>, where <value> is the relative weight of the field.

   The following fields have elevated scoring weights by default:

   - kb_knowledge.number = 50
   - kb_knowledge.short_description = 10
   - kb_knowledge.meta = 10
   - task.number = 50
   - task.short_description = 10

   For example, if the Title field has a ts_weight=50 and the Description field has a ts_weight=10, then a match in the title is weighted 5 times more relevant than a match in the description and 50 times more relevant than a field without the ts_weight attribute.

3. Click Update.

   Note: To view the complete scoring system, see Document Scoring.
Zing indexes words

Index documents by dividing them into words. Depending on the languages your instance supports, a word may be a single character such as a Chinese or Japanese pictogram or a sequence of characters separated by spaces such as with Latin, Arabic, and Pinyin languages.

Zing uses spaces to define the basic unit of word separation. Zing uses the following rules to index words.

**Table 283: Zing word indexing rules**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punctuation</td>
<td>Zing indexes some punctuation marks as part of some words to improve search results for common search terms. Zing converts any non-indexed punctuation characters to spaces. See Zing indexes punctuation as part of some words on page 1094.</td>
</tr>
<tr>
<td>Pictograms</td>
<td>Zing supports the full range of Unicode pictogram characters. Each Chinese or Japanese pictogram is indexed as a separate word as if it were a single Latin-1 character surrounded by spaces.</td>
</tr>
<tr>
<td>Letter</td>
<td>Zing treats Latin-1, Arabic, and Pinyin characters as individual letters of space-separated-words. Sequences of letters define indexable words.</td>
</tr>
</tbody>
</table>

*Zing indexes punctuation as part of some words*

Zing indexes some punctuation marks as part of some words to improve search results for common search terms.

**Table 284: Punctuation indexed as part of a word**

<table>
<thead>
<tr>
<th>Punctuation</th>
<th>Indexed to find</th>
<th>Search behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ampersands</td>
<td>Company names</td>
<td>Zing locates terms containing punctuation characters that are common in company names: ampersands (&amp;), plus signs (+), or hyphens (-). For example, a search for H&amp;R Block locates exact matches to the search term.</td>
</tr>
<tr>
<td>Apostrophes</td>
<td>Proper names</td>
<td>Zing ignores trailing possessives and retains most others to find company names. For example, a search for O'Reilly's locates matches to O'Reilly and O'Reilly's.</td>
</tr>
<tr>
<td>Punctuation</td>
<td>Indexed to find</td>
<td>Search behavior</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>Hyphens [-]</strong></td>
<td>Company names</td>
<td>Zing locates terms containing punctuation characters that are common in company names: ampersands (&amp;), plus signs (+), or hyphens (-). For example, a search for <em>Coca-Cola</em> locates exact matches to the search term.</td>
</tr>
<tr>
<td><strong>Product numbers</strong></td>
<td></td>
<td>Zing locates product numbers that follow a typical pattern. To meet this pattern, search terms must contain frequent numbers and only include underscores (_), hyphens (-), or periods for punctuation. For example, a search for <em>PROD-10-987</em> locates exact matches to the search term.</td>
</tr>
<tr>
<td><strong>Numbers [123]</strong></td>
<td>Product numbers</td>
<td>Zing locates product numbers that follow a typical pattern. To meet this pattern, search terms must contain frequent numbers and only include underscores (_), hyphens (-), or periods for punctuation. For example, a search for <em>PROD10987</em> locates exact matches to the search term.</td>
</tr>
<tr>
<td><strong>Record numbers</strong></td>
<td></td>
<td>Zing locates record numbers that follow a typical pattern. To meet this pattern, search terms must start with a recognized record number prefix. For example, a search for <em>INT1234567</em> locates exact matches to the Incident record.</td>
</tr>
<tr>
<td><strong>Periods [.]</strong></td>
<td>Acronyms</td>
<td>Zing locates acronyms whether they are separated by periods or not. For example, a search for <em>u.s.a.</em> locates matches to <em>usa</em> or <em>u.s.a.</em>. Note that wildcard searches may affect acronym handling. For example, a search for <em>u.s.</em> may yield better results than a search for <em>u.s.</em>.</td>
</tr>
<tr>
<td>Punctuation</td>
<td>Indexed to find</td>
<td>Search behavior</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>Host names</strong></td>
<td>Zing locates sequences of letters and numbers separated only by periods. For example, a search for en.myhost123.com locates exact matches to the search term.</td>
<td></td>
</tr>
<tr>
<td><strong>IP addresses</strong></td>
<td>Zing locates numbers that follow a typical IPv4 address pattern. For example, a search for 10.0.0.1 locates exact matches to the search term.</td>
<td></td>
</tr>
<tr>
<td><strong>Product numbers</strong></td>
<td>Zing locates product numbers that follow a typical pattern. To meet this pattern, search terms must contain frequent numbers and only include underscores (_), hyphens (-), or periods for punctuation. For example, a search for PROD.10.987 locates exact matches to the search term.</td>
<td></td>
</tr>
<tr>
<td><strong>Plus signs [+]</strong></td>
<td>Company names</td>
<td>Zing locates terms containing punctuation characters that are common in company names: ampersands (&amp;), plus signs (+), or hyphens (-). For example, a search for Google+ locates exact matches to the search term.</td>
</tr>
<tr>
<td><strong>Underscores [_]</strong></td>
<td>Product numbers</td>
<td>Zing locates product numbers that follow a typical pattern. To meet this pattern, search terms must contain frequent numbers and only include underscores (_), hyphens (-), or periods for punctuation. For example, a search for PROD_10_987 locates exact matches to the search term.</td>
</tr>
</tbody>
</table>

**Note:** The punctuation handling described here does not affect wildcard and boolean operators. These operators provide separate Zing functions.

*Zing indexes some HTML elements*

Zing indexes the contents of certain HTML elements to improve document search results. This indexing is addition to the normal indexing by word.
Geneva ServiceNow ServiceNow Platform

Table 285: Indexed HTML elements

<table>
<thead>
<tr>
<th>HTML element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title attribute</td>
<td>Zing indexes the contents of the title attribute of any HTML element. This is separate from the title element, which Zing already indexes when identifying the words of the document.</td>
</tr>
<tr>
<td>Anchor link target</td>
<td>Zing indexes the target URL specified in the href attribute of any anchor element.</td>
</tr>
<tr>
<td>Alt text for an image</td>
<td>Zing indexes the alternative text of any image element.</td>
</tr>
</tbody>
</table>

*Enable text indexing for a table*
Administrators can enable text indexing on a table to allow users to search for string values from table records.

Role required: admin

By default, the system creates text indexes for the tables with a Text index record (*System Definition > Text Indexes*). For example:

- Knowledge-related tables
- Core-data-related tables
- Connect-related tables
- Task table
- User table

Text indexing can be a resource-intensive task that may take a while to complete. You may notice performance degradation or incomplete search results during index generation. To estimate text indexing duration, you can view historical statistics.

1. Navigate to *System Definition > Dictionary*.
2. Click the table name to open the dictionary entry.
   
   A table in the System Dictionary is a record that has a Table name but no Column name.
3. Select the **Text index** check box to enable text indexing for text fields on the table.

4. Click **Update**.
5. Click **Generate Text Index**.

The system schedules the table for text indexing, typically within a minute.

The Search for text option appears on the list view for the table.

**Regenerate a text index for a table**

You can regenerate a table text index when you change table stop words or display values.

Role required: admin

By default, the system maintains text indexes on a daily schedule. Typically, you only need to manually regenerate a text index when you change these values.

- You change the list of table-specific stop words.
- You change the display value of a record such as changing a user or group name.
Until you regenerate the index, text searches for old display values will still produce results and searches for the new display value will not show results.

Text indexing can be a resource-intensive task that may take a while to complete. You may notice performance degradation or incomplete search results during index generation. To estimate text indexing duration, you can view historical statistics.

1. Navigate to **System Definition > Text Indexes**.
2. Open the text index for the table. For example, select task.
   The system displays the Text Index record for the table.
3. Click the **Regenerate Text Index** related link and click **OK**.
<table>
<thead>
<tr>
<th>Table</th>
<th>Task [task]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>12</td>
</tr>
<tr>
<td>State</td>
<td>Ready</td>
</tr>
<tr>
<td>Indexed rows</td>
<td>463</td>
</tr>
<tr>
<td>Unique terms</td>
<td></td>
</tr>
<tr>
<td>Total terms</td>
<td></td>
</tr>
<tr>
<td>Disable synonym</td>
<td></td>
</tr>
<tr>
<td>Partial Match Rule</td>
<td></td>
</tr>
</tbody>
</table>

**Generate Text Index**

The text index will be generated in the background so that you can continue other work while it is being created. If this table does not have a text index, one will be created after the dictionary text index flag is set to true. Upon completion, the system will send you a confirmation email unless you specify otherwise below.

Upon completion:
- **Email me**
  - **admin@example.com**
- **Do not notify me**

**Text Index Specifics**

- **Table name**: task
- **Task**: task

**Index Stop Words**

- **Index Stop Words**
  - New
  - Go to
  - Word
  - Search

**Index = task**

- **Index Stop Words**
- **Index Synonym Dictionaries**

No words to display.
The system schedules the table for text indexing.

**Regenerate a corrupt index**

Typically, a corrupt index regenerates itself. If the system does not start the regeneration process, you must issue a script command to delete the existing index and regenerate it.

For example, to regenerate the task text index, the following script could be used as a background script:

```java
var indexer = new Packages.com.glide.lucene.TextIndexEvent();
indexer.createIndex("task", "my_email@service-now.com");
```

Change the email address to the address that receives the notification when the index is completed. The script creates an event that the index handler processes so the script completes almost instantly. When the scheduled job picks up the event, you can check the Component Status page and see that the index is being generated. An entry also appears in the log, as shown in the following example:

```
10/28/09 15:36:19 (521) worker.1 worker.1 TextIndexGenerator: Text index generation starting for: task
```

As each extended table completes, it is logged:

```
10/28/09 15:37:13 (213) worker.1 worker.1 [0:00:01.462] TextIndex generation complete for: problem, rows indexed: 11
```

After all the extended tables are completed, the Component Status page entry is updated to "completed" and a final log entry is made. The log entry shows how long the process took.

```
10/28/09 15:37:20 (794) worker.1 worker.1 TextIndexGenerator: Text index generation complete for: task, in: 0:01:02.669
```

**Remove an index**

You can remove an index if you no longer want the search engine to return results for table.

1. Navigate to **System Definition > Dictionary**.
2. Open the dictionary entry for the table.
3. Clear the **Text index** check box and click **Update**.
   
   The system no longer indexes or queries the table for text search results.

**Remove an index for a specific field**

You can remove an index if you no longer want the search engine to return results for a specific field.

1. Navigate to a form containing the field.
2. Right-click the field label and select **Configure Dictionary**.
3. Add `no_text_index=true` to the **Attributes** field. Separate multiple attributes with a comma (with no spaces).
4. Click **Update**.
   
   The system no longer indexes or queries the field for text search results.

**Text indexing statistics and status**

To view text indexing statistics and status, navigate to **System Definition > Text Indexes**.

- If text indexing is in progress, view the status of each index in the **State** field. Refresh the list to view updates. Tables are indexed one at a time.
- To see how long it takes to index a specific table, view the **Last indexing duration** field for the table entry. Although each process varies based on activity and data, historical data can provide a good estimate.
• View statistics (such as the number of indexed rows, terms, and queries in the past week) in the list or on the form for a specific table.

Figure 368: Text Index Status

Zing can include attachments in search results

Expand search results to include attachments from indexed tables.
By default, only searches of Knowledge Base records include attachments. Administrators can enable searching attachments for other tables, but doing so causes the platform to re-index the selected table, its parent table, and any children of the parent table.

**Warning:** For large tables, such as the Task table, re-indexing can take several hours and slows down the system until complete. Re-indexing is best performed during non-peak times.

Zing supports indexing and searching these attachment file types:

- .doc
- .htm
- .html
- .ini
- .pdf
- .ppt
- .reg
- .txt
- .xls
- .docx
- .dotx
- .dot
- .pptx
- .potx
- .pot
- .xltx
- .xlt
Enable attachment indexing on a table
When you enable the attachment indexing for a table, text searches can return matches from the record and its file attachments.

Role required: admin

By default, attachment indexing is enabled for the Knowledge Base.

Enabling attachment indexing causes the platform to re-index the selected table, its parent table, and any children of the parent table.

Warning: For large tables, such as the Task table, re-indexing can take several hours and slows down the system until complete. Re-indexing is best performed during non-peak times.

1. Navigate to System Definition > Dictionary and select the record for the table.
2. Click the Attributes tab.
3. Click New and add the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Attachment index</td>
</tr>
<tr>
<td>Value</td>
<td>true</td>
</tr>
</tbody>
</table>

Note: The attachment indexing attribute only applies to the tables on which you explicitly add it. It does not cascade to child tables. For example, indexing attachments on the Task table does not index attachments on the Incident table.

4. Click Submit.
5. Optional: To disable attachment indexing, remove the attribute.
6. Click Update.
   The indexing process begins. When it is complete, attachments can be searched on that table.

Zing removes stop words from queries
Remove common words from search queries that do not produce meaningful results.

By default, the system maintains two types of stop words.

<table>
<thead>
<tr>
<th>Stop word type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System-wide text index stop words</td>
<td>The system always ignores system-wide text index stop words when generating text indexes. Any search for a system-wide stop word returns no search results.</td>
</tr>
<tr>
<td>Table-specific stop words</td>
<td>The system uses the table-specific Text Index record to determine whether to index the stop word or to just remove it from keyword search queries against the table.</td>
</tr>
</tbody>
</table>
By default, the system has stop words for common English words. Search administrators typically create stop words from search terms that produce too many search results such as articles, conjunctions, personal pronouns, and prepositions.

Configure a global stop word
Configure stop words that should not be indexed by the search.

1. Navigate to **System Definition > Text Index Stop Words**.
2. Add or remove stop words from the list.
3. If a message appears at the top of the list, contact Technical Support to regenerate all indexes.
   You must regenerate indexes whenever words may be missing from an index. For example, if you delete, inactivate, or change an active global stop word, the word may be missing from the index. An after business rule checks these conditions and generates the notification message when index regeneration is necessary.

Configure a table-specific stop word
You can configure stop words for a specific table.

1. Navigate to **System Definition > Text Indexes**.
2. Open the text index entry for the table.
3. Add or remove stop words from the **Index Stop Words** related list.
   For each word, select the desired **Stop mode** (Not a Stop Word (inactivates the stop word), Neither Index nor Query, or Index but do not Query).
4. If a message appears at the top of the form, click **Regenerate Text Index** and click **OK**.
   You must regenerate indexes whenever words may be missing from an index. For example, if you delete or change a stop word for which the mode was Neither Index nor Query, the word may be missing from the index. An after **business rule** checks these conditions and generates the notification message when index regeneration is necessary.

**Note:** Text search uses the global list plus the table-specific list of stop words when indexing the table.
Enable automatic stop words for a table
The system can identify and generate stop words when a search term exceeds an occurrence threshold.

- Role required: admin
- Enable text indexing for the table

By default, the TS Index Stats scheduled job identifies and creates stop words for tables with text indexes on a nightly basis. Automatically adding stop words improves your search results by removing search terms that return too many search results.

1. Navigate to System Definition > Text Indexes.
2. Open the text index entry for the table.
3. Select the Auto stop check box.
4. In Auto threshold, enter the maximum number of occurrences a search term can have in search results.
   
   When a search term generates more search results than the threshold, the system automatically creates a stop word for the search term.
For example, to create stop words for the task table, see the blog post *Configuring auto stop words and regenerating text indexes* from a ServiceNow employee on the ServiceNow Community.

5. Click **Update**.

The job identifies and creates stop words for the table with a **Stop mode** value of *Index but do not Query* and inserts a **Comment** to indicate that the stop word was generated automatically.

**Zing matches derived words with stemming**

Convert any multiple-character search keyword to its stem form to find derived versions of the word.

Stemming removes a variety of common word inflections, such as plurals and past tense forms. Zing treats all words with a common stem as synonyms of the original search term. Stemming does not apply to searches involving single-character words such as Chinese and Japanese pictograms.

Zing uses the *Porter Stemming Algorithm*, which is most effective for English text, but supports stemming in these languages.

- English
- German

The stemming language determines how the system generates indexes of record data. The system only supports one stemming language at a time regardless of how many languages the instance supports. When you change the stemming language, the system creates any new index entries based on the new language stemming rules, but does not regenerate any existing index records. You must manually regenerate indexes for tables you want to use the new stemming language.

**Installed with Zing**

Several types of components are installed with Zing.

**Table 288: Tables installed with Zing**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Index [ts_index_name]</td>
<td>Stores the tables the system indexes.</td>
</tr>
<tr>
<td>Index Stop Word [ts_index_stop]</td>
<td>Stores the stop words for a specific table.</td>
</tr>
<tr>
<td>Stop Word [ts_stop]</td>
<td>Stores the global stop words.</td>
</tr>
<tr>
<td>Text Search Groups [ts_group]</td>
<td>Stores search groups for global text search.</td>
</tr>
<tr>
<td>• ts_attachment</td>
<td>System tables that support Zing. Extending or</td>
</tr>
<tr>
<td>• ts_chain_summary</td>
<td>modifying these tables is not recommended.</td>
</tr>
<tr>
<td>• ts_chain</td>
<td></td>
</tr>
<tr>
<td>• ts_deleted_doc</td>
<td></td>
</tr>
<tr>
<td>• ts_document</td>
<td></td>
</tr>
<tr>
<td>• ts_index_stats</td>
<td></td>
</tr>
<tr>
<td>• ts_phrase</td>
<td></td>
</tr>
<tr>
<td>• ts_search_stats</td>
<td></td>
</tr>
<tr>
<td>• ts_search_summary</td>
<td></td>
</tr>
<tr>
<td>• ts_word_summary</td>
<td></td>
</tr>
<tr>
<td>• ts_word_roots</td>
<td></td>
</tr>
<tr>
<td>• ts_word</td>
<td></td>
</tr>
</tbody>
</table>
### Table 289: Business rules installed with Zing

<table>
<thead>
<tr>
<th>Business Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Search Property Change Rationally</td>
<td>Ensures that valid values are entered for Zing text search properties.</td>
</tr>
<tr>
<td>Text Index Stop Reminder</td>
<td>Warns the user of stop word changes that require the index to be rebuilt (table-specific). The warning is issued if record is deleted that had a stop mode “neither index nor query”, if record's stop mode is updated to something else and was “neither index nor query”, and if record's word is updated to something else and stop mode is “neither index nor query”.</td>
</tr>
<tr>
<td>Stop Word Reminder</td>
<td>Warns the user of stop word changes that require the index to be rebuilt (global). The warning is issued if record is deleted and it was active, if record is inactivated, and if record's word is changed and was active.</td>
</tr>
</tbody>
</table>

### Table 290: Scheduled jobs installed with Zing

<table>
<thead>
<tr>
<th>Scheduled job</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS Search Stats</td>
<td>Compiles type-ahead suggestions each night. See <a href="#">Update A Type-Ahead Suggestion</a>.</td>
</tr>
<tr>
<td>TS Index Stats</td>
<td>Collects statistics and performs maintenance for text search and indexing. Runs nightly.</td>
</tr>
<tr>
<td>text index events process</td>
<td>Collects statistics and performs maintenance for text search and indexing. Runs every 30 seconds.</td>
</tr>
<tr>
<td>TS Search Summary</td>
<td>Compiles top search statistics each hour. See <a href="#">Update a top search statistic</a> on page 483.</td>
</tr>
<tr>
<td>TS Chain Summary</td>
<td>Compiles search chain statistics each hour.</td>
</tr>
</tbody>
</table>

### Table 291: UI action installed with Zing

<table>
<thead>
<tr>
<th>UI action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regenerate Text Index</td>
<td>Displays the <a href="#">Regenerate Text Index</a> link on Text Index forms.</td>
</tr>
</tbody>
</table>

### Table 292: Homepage installed with Zing

<table>
<thead>
<tr>
<th>Homepage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Searches homepage</td>
<td>Displays the most popular searches over the past hour, day, week, or month.</td>
</tr>
</tbody>
</table>
Google custom search integration

Replace Zing text indexing and search engine with Google Site Search functionality.

Activating custom search integration replaces the default search and provides Google Site Search functionality. You can configure your Google Site Search engine to include or exclude sites in your search results. For more information on the Google Site Search feature, see Google Site Search.

Activate Google custom search integration

The Google Custom Search Integration plugin is available by request.

Role required: none

Request the plugin through the HI Service Portal.

1. In the HI Service Portal, click Service Catalog > Activate Plugin.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
<tr>
<td>Specify the date and time you would like this plugin to be enabled</td>
<td>Date and time must be at least 2 business days from the current time.</td>
</tr>
</tbody>
</table>

**Note:** Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.

| Reason/Comments | Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows. |

3. Click Submit.

Configure a search property

The custom search properties allow you to specify the Google Site Search key and the name of the page that appears on the search form. You can also specify whether to open search results in a new page.

Role required: admin

1. Navigate to Custom Search Integration > Properties.
2. Edit the properties.
### Table 293: Search property table

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Google Custom Search key [gc_search.key] | The Google custom search key from the Google site.  
  - Type: String  
  - Default value: None |
| Name of your search to appear on form [gc_search.name] |  
  - Type: String  
  - Default value: None |
  - Type: boolean  
  - Default value: Yes |

3. Click **Save**.

### Define a custom search page

Users can access the custom search by navigating to [https://<your_instance_name>]/gc_search_widget.do to directly access the Google Site Search page. Administrators can also define custom search pages for an instance.

**Role required:** admin

1. To create the search page, complete the following steps.
   a) Navigate to **System UI > UI Pages**.
   b) Create a custom **UI page**.
   c) Embed the **gc_search_widget UI page** in the new search page.

2. Navigate to **Custom Search Integration > Pages**.
3. Click **New**.
4. Complete the form.

### Table 294: Search page table

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>The URL of the search page.</td>
</tr>
<tr>
<td>Title</td>
<td>The title to display at the top of the search page.</td>
</tr>
<tr>
<td>URL Domain</td>
<td>The domain of the search page.</td>
</tr>
<tr>
<td>Weight</td>
<td>The weight assigned to the page.</td>
</tr>
<tr>
<td>Average user weighting</td>
<td>The average user weighting for the page.</td>
</tr>
</tbody>
</table>
5. Click Submit.

Set up the Google Enterprise Search

To configure the Google Site Search engine on your instance, you must first go to the Google website and obtain a Search Engine ID, set up the Google Enterprise Search, and then register the ID with your instance.

1. Go to Google Site Search and sign up for Google Site Search. You must sign up for a paid version of Google Site Search; the free version is not supported. Your organization must be compliant with the Google Site Search licensing terms.

2. Set up your Google Site Search by going to the Custom Search page while logged into your Enterprise account. In the Custom Search definition, you can define the sites to search and the sites to exclude. Click Search engine ID to retrieve your Google Custom Search Key, which you need to configure the search on your ServiceNow instance.

3. Add the URLs you want your users to be able to search. To add a website, click Add under Sites to search.

4. To exclude sites, click Advanced.
5. In your ServiceNow instance, go to **Custom Search Integration > Properties**, enter the **Google Custom Search key** that you created on the Google web site, and click **Save**.

![Figure 372: Configuring Google search engine ID](image)

**Block a robot from your site**

Your instance is configured by default to disable robots from crawling your site. However, if you choose to use the Google Site Search engine, your site will be crawled by the Google robots unless you specifically configure it not to do so.

Role required: admin

The Robots.txt Definition module allows you to define the contents of the robots.txt file associated with the instance. The syntax used is the same as for a standard robots.txt file. For more information on how to format the robots.txt file, see **Robots.txt Specifications**.

1. Go to **Custom Search Integration > Robots.txt Definitions**.

2. Click **New**.
3. Enter the contents of the robots.txt file in the Text field. If you want this file to be active, select the **Active** field. There can only be one robots.txt file active at any time. Setting a file to active automatically sets all other files to inactive. Web crawlers will recognize the contents of the active robots.txt file and honor the robots exclusion protocol.

4. Click **Submit**.

**Retaining only the web crawling blocker**

If you only want to configure web crawlers using the robots.txt file, and not deploy the Google Site Search engine on your instance, you can request the Remove Google Custom Search update set from ServiceNow support.

This update set removes the Google search functionality but leaves the web crawler configuration capability.

**Viewing the search history**

The search history shows all of the searches made on the instance in chronological order.

To view the search history, go to **Custom Search Integration > Search History**. To delete an entry, click the query name and click **Delete**.
Table 295: Search history

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>The text of the search query.</td>
</tr>
<tr>
<td>User</td>
<td>The user who executed the search.</td>
</tr>
</tbody>
</table>

Figure 373: Google search history

Mobile device configuration

Take the system on the go with mobile device support.

Get started with a ServiceNow instance on your mobile device

Use your mobile device to access an instance in a web browser or the native app.

Before you begin

You can access your instance from a web browser on your mobile device or from the native mobile app. Download the native app from the iTunes store or from the Google Play store. Make sure you are on an instance using Geneva patch 6 or higher to use the Android app. If you do not have access to the mobile platform, contact your administrator.
What to do

Access an instance on your mobile device  
To open an instance on a mobile device:
• open an instance in your mobile app  
• open an instance in your mobile browser

For mobile web access, add a shortcut to an instance browser page on your mobile device.

Set up homepage favorites  
Use the application navigator to find modules and applications.
Create and edit favorite modules or applications to appear as icons on your homescreen.

Use lists and records on a mobile device  
View a list on your mobile device by tapping a homescreen favorite or by opening a module from the application navigator.
View a recently accessed list.
View a recently accessed record.
Add a tag to a record.
View the related list for a record.

Use Service Catalog, Connect Chat, location, and barcode scanning  
Use Service Catalog to order materials.
Collaborate and stay connected with other users using Connect Chat.
Allow the ServiceNow app to access your location and camera to check in or scan barcodes.

Use an instance on a tablet web browser  
When you access an instance from a tablet browser, your tablet is automatically detected.
Most features that are available on a desktop instance are also available in the tablet browser.
Several features have been optimized for the tablet interface.
Access modules and applications in the application navigator.
Access the following tablet context menus:
• List header menu
• List row menu
• Form menu
Filter search results further by searching in a column header.
Group and organize records, and find them more easily by adding labels to the record.

Next steps

The mobile UI varies depending on the device you are using and whether you are accessing your instance from the native app or from a mobile browser. Use this section to learn more about the mobile UI.
Mobile app UI

Access an instance from the mobile interface using a native mobile app.

Download the mobile app from the iTunes store for devices running iOS 9 and above or from the Google Play store for Android phones running KitKat and above.

Supported devices for the mobile app

Certain devices and operating systems are supported by the mobile app.

<table>
<thead>
<tr>
<th>Device</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple® iPhone®</td>
<td>All models of iPhone running iOS 9 and above.</td>
</tr>
<tr>
<td>Apple® iPod® touch</td>
<td>All models of iPod running iOS 9 and above.</td>
</tr>
<tr>
<td>Apple® iPad®</td>
<td>All models of iPad running iOS 9 and above.</td>
</tr>
<tr>
<td>Android™ mobile devices</td>
<td>All models of Android phone running 4.4 (KitKat) and above.</td>
</tr>
<tr>
<td></td>
<td>Only Geneva patch 6 and higher is supported on the Android app.</td>
</tr>
</tbody>
</table>

Features with limited mobile support

Some features are not supported in the native mobile app.

**Not supported**

- E-signature
- Knowledge v2
- Connect Support

**Limited support**

- List filtering: Several fields listed below have limited support on mobile devices. You can still create a complex filter in a desktop instance and open it in the mobile app. However, you won't be able to edit any of the limited fields.
  - between
  - dates
  - tags
  - related fields
- Two factor authentication is not supported, however single sign on and basic authentication are supported.

Access an instance from the mobile app

You can access an instance using the mobile app for supported Apple devices.

To use the Android mobile app, you must be on an instance using Geneva patch 6 or higher.

1. Download the ServiceNow application from the Apple App Store or from the Google Play store.
2. Open the app by tapping the icon on the home page. The launch screen opens.

3. In the Instance Name field, enter the name of your instance. You don't need to include service-now.com at the end of the instance name.

4. Tap Go. The login screen appears.
5. Use your regular ServiceNow login to sign in to the app.

After you log in, the app remembers your information. As long as you don't log out, the app opens the last screen you used when you open it again.

If you do log out, the app remembers your instance. You can select the instance from the History list to log back in.
iOS and Android UI comparison

The native mobile app has the same functionality across platforms.

Android and iOS mobile apps are similar with a few minor differences to make the experiences unique to each platform. Any differences between the two platforms are designed to make the app accessible to a user on either device. For example, the location of the Navigation Menu varies between the two platforms to stay consistent with the platform-specific UI.

To use the Android mobile app, you must be on an instance using Geneva patch 6 or higher.
iOS app UI

Android app UI

Good Afternoon,
System Administrator

Active
Software 7
Hardware 4
Issues 13

Create New

Knowledge

Critical incident

19

My Approvals
My Incidents

Location
Connect

Active
Create New
Knowledge

My Approvals
My Incidents
Critical incidents

Location
Connect

© 2017 ServiceNow. All rights reserved.
Mobile app UI home

From the homepage, access favored items, the application navigator, and your profile.

Favorites are the basis of the mobile homepage. Your homepage appearance depends on what favorites you have in your instance.
### Mobile UI home iPhone

#### Good Evening, System Administrator

- **Active**
- **Create New**
- **My Approvals**
- **My Incidents**
- **Location**
- **Connect**

### Mobile UI home Android

#### Good Evening, System Administrator

- **Active**
- **Create New**
- **My Approvals**
- **My Incidents**
- **Location**
- **Connect**
### Table 297: Elements of the mobile UI home

<table>
<thead>
<tr>
<th>Number</th>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User profile button</td>
<td>Displays the current profile record.</td>
</tr>
<tr>
<td>2</td>
<td>Favorites</td>
<td>Links to records in the instance, sometimes with visualizations.</td>
</tr>
<tr>
<td>3</td>
<td>Application navigator button</td>
<td>Displays the list of menus and modules available to the current user.</td>
</tr>
</tbody>
</table>

### Mobile favorites

Mobile favorites provide links to records in the system. The mobile app displays favorites as an icon on the home page.

Users can create favorites for the following interface elements.

- List filters
- Records
Figure 374: Sample mobile UI home page with favorites

Users can tap and hold a favorite to edit the following properties.

- Label
- Text and icon color
- Icon image
- Enable visualizations

The instance automatically synchronizes favorites between the mobile UI and the desktop UI.

Visualizations

Visualizations allow users on the mobile app UI to see relevant table information directly from the home page favorite icon.

The mobile UI offers the following visualizations.
Table 298: Favorite visualization options

<table>
<thead>
<tr>
<th>Visualization option</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>The favorite icon displays the number of records returned by the list filter.</td>
<td><img src="assigned-to-me.png" alt="Count Example" /></td>
</tr>
<tr>
<td>Bar chart</td>
<td>The favorite icon displays the number of records with a particular field value. You select the field whose values you want to display when you set up the visualization.</td>
<td><img src="assigned-to-me.png" alt="Bar Chart Example" /></td>
</tr>
</tbody>
</table>

Use the application navigator

Use the application navigator to access different modules in the mobile interface.

To open the application navigator, tap the menu icon (for iPhone and for Android).
Scroll the navigator up or down using one or two fingers. To expand an application, tap the application name. To hide the navigator, tap **Close** or the back arrow.
Mobile lists

Lists appear as a single column of records on the mobile app UI display. Each row represents a separate record.
<table>
<thead>
<tr>
<th>Mobile UI lists for iPhone</th>
<th>Mobile UI lists for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network storage unavailable</strong></td>
<td><strong>Unable to get to network file shares</strong></td>
</tr>
<tr>
<td>INC00000049</td>
<td>INC0000002</td>
</tr>
<tr>
<td><strong>My desk phone does not work</strong></td>
<td><strong>Wireless access is down in my area</strong></td>
</tr>
<tr>
<td>INC00000041</td>
<td>INC0000003</td>
</tr>
<tr>
<td><strong>Rain is leaking on main DNS Server</strong></td>
<td><strong>Need access to sales DB for the West</strong></td>
</tr>
<tr>
<td>INC00000016</td>
<td>INC0000007</td>
</tr>
<tr>
<td><strong>JavaScript error on hiring page of corporate website</strong></td>
<td><strong>I can't launch my VPN client since the last software update</strong></td>
</tr>
<tr>
<td>INC00000040</td>
<td>INC0000015</td>
</tr>
<tr>
<td><strong>Can't access SFA software</strong></td>
<td><strong>Rain is leaking on main DNS Server</strong></td>
</tr>
<tr>
<td>INC00000046</td>
<td>INC0000016</td>
</tr>
<tr>
<td><strong>SAP Financial Accounting application appears to be down</strong></td>
<td><strong>How do I create a sub-folder</strong></td>
</tr>
<tr>
<td></td>
<td>INC0000017</td>
</tr>
</tbody>
</table>
Mobile lists consist of the following elements.

Table 299: Elements of the mobile UI: Lists

<table>
<thead>
<tr>
<th>Number</th>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>List name</td>
<td>Displays the name of the list favorite.</td>
</tr>
<tr>
<td>2</td>
<td>Back button</td>
<td>Navigates back to the home page.</td>
</tr>
<tr>
<td>3</td>
<td>UI actions</td>
<td>Displays buttons to display the list activity stream for iPhone and add a record.</td>
</tr>
<tr>
<td>4</td>
<td>Application search</td>
<td>Displays records that match the search string. On Android, search is located in the List options menu.</td>
</tr>
<tr>
<td>5</td>
<td>Current filter</td>
<td>Displays the conditions filtering the list.</td>
</tr>
<tr>
<td>6</td>
<td>Records</td>
<td>Displays one row for each record in the list.</td>
</tr>
<tr>
<td>7</td>
<td>List options</td>
<td>Displays list options to add to favorites, share, and sort the list.</td>
</tr>
</tbody>
</table>

Mobile activity streams
List activity streams appear as a single column window with a separate row for each record update in the mobile UI.

Access a list activity stream by pressing the activity stream icon. Activity streams open on a new page. Use the List options menu to open the Activity Stream on Android.
Figure 375: iPhone list activity stream
Figure 376: Android list activity stream

Figure 377: Elements of the mobile UI: activity stream
Activity stream updates consist of the following elements.

Table 300: Mobile UI elements: activity stream

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back button</td>
<td>Return to the list.</td>
</tr>
<tr>
<td>User</td>
<td>Displays the user photo and name of the user who made the activity update.</td>
</tr>
<tr>
<td>Record details</td>
<td>Displays the record number and relative time since the update was made.</td>
</tr>
<tr>
<td>Activity details</td>
<td>Displays the most recent activity update.</td>
</tr>
</tbody>
</table>

Swipe up or down to see more activity stream updates.
**Mobile filters**

Construct complex filters with the mobile app UI.

<table>
<thead>
<tr>
<th>Number</th>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Current set</td>
<td>Displays the current condition set and the number of records returned by the condition set.</td>
</tr>
<tr>
<td>Number</td>
<td>UI element</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2</td>
<td>Current rule</td>
<td>Displays the type of rule set that applies to the current filter.</td>
</tr>
<tr>
<td>3</td>
<td>Conditions</td>
<td>Displays the conditions that are part of the current condition set.</td>
</tr>
<tr>
<td>4</td>
<td>Add rule button</td>
<td>Displays a pop-up to add a rule.</td>
</tr>
<tr>
<td>5</td>
<td>New filter button</td>
<td>Creates a condition set.</td>
</tr>
<tr>
<td>6</td>
<td>Delete set</td>
<td>Deletes the current condition set.</td>
</tr>
</tbody>
</table>

Not all filters are available on the mobile app.

Mobile condition sets

A condition set is an optional feature to generate a list by combining the results of multiple sets of conditions. Typically, condition sets are used to create complex filters.

Each set specifies whether a record must match all conditions or whether a record can match any condition. In general, condition sets requiring matching to all conditions return fewer records than sets that allow matching on any condition within a set.

For example, consider the following filter consisting of two condition sets.

**Figure 378: Sample list filter with two condition sets**
In this example, the first condition set requires matching all the conditions and returns 8 results, whereas the second set allows matching to any condition and returns 29 results. The total results count of 37 records is the combination of record results from the individual condition sets.

Mobile service catalog
The mobile app UI provides a native mobile interface for browsing the service catalog.
The mobile service catalog user interface consists of the following elements.

Table 302: Elements of the mobile UI: service catalog

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current catalog</td>
<td>Displays the title of the current service catalog.</td>
</tr>
<tr>
<td>Back button</td>
<td>Returns to the previous screen.</td>
</tr>
<tr>
<td>Search</td>
<td>Enter search strings for categories or catalog items.</td>
</tr>
<tr>
<td>UI element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Categories</td>
<td>Search for catalog items by a logical grouping.</td>
</tr>
<tr>
<td>Application navigator button</td>
<td>Displays the list of menus and modules available to the current user.</td>
</tr>
<tr>
<td>Shopping cart</td>
<td>Displays the shopping cart to check out.</td>
</tr>
<tr>
<td>Share button</td>
<td>Displays the document providers available to share the current record.</td>
</tr>
</tbody>
</table>

**Mobile service catalog categories**

The mobile app UI provides a native mobile interface for browsing service catalog categories. The mobile app UI displays each category in its own page.
Figure 380: Elements of the mobile UI: service catalog category

Mobile service catalog categories consist of the following elements.

Table 303: Elements of the mobile UI: service catalog category

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current category</td>
<td>Displays the currently selected service catalog category.</td>
</tr>
<tr>
<td>Back button</td>
<td>Returns to the previous service catalog page.</td>
</tr>
<tr>
<td>UI element</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Search</td>
<td>Enter search strings for categories or catalog items.</td>
</tr>
<tr>
<td>Category filter</td>
<td>Displays the current category filter.</td>
</tr>
<tr>
<td>Catalog items</td>
<td>Displays one row for each catalog item.</td>
</tr>
<tr>
<td>Shopping cart</td>
<td>Displays the shopping cart to check out.</td>
</tr>
</tbody>
</table>

The category filter allows mobile users to switch between categories and subcategories.

![Category Filter Diagram]

Figure 381: Elements of the mobile UI: category filter

The category filter consists of the following elements.
### Table 304: Elements of the mobile UI: category filter

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel button</td>
<td>Closes the category filter.</td>
</tr>
<tr>
<td>Current category</td>
<td>Displays the currently selected category.</td>
</tr>
<tr>
<td>Current filter</td>
<td>Displays the current filter applied to categories and subcategories.</td>
</tr>
<tr>
<td>Subcategories</td>
<td>Displays the subcategories available to filter the service catalog.</td>
</tr>
<tr>
<td>Apply button</td>
<td>Saves changes to the category filter.</td>
</tr>
</tbody>
</table>

*Mobile catalog items*

The mobile app UI provides a native mobile interface for browsing service catalog items.
Figure 382: Elements of the mobile UI: catalog item

The mobile interface for a mobile catalog item consists of the following elements.

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current item</td>
<td>Displays the name of the currently selected catalog item.</td>
</tr>
<tr>
<td>Back button</td>
<td>Returns to the previous screen.</td>
</tr>
<tr>
<td>Item description</td>
<td>Displays the catalog item description.</td>
</tr>
</tbody>
</table>

Table 305: Elements of the mobile UI: catalog item
<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item variables</td>
<td>Displays input fields appropriate to the catalog variable type.</td>
</tr>
<tr>
<td>Add to cart button</td>
<td>Adds the current item to shopping cart.</td>
</tr>
<tr>
<td>Shopping cart</td>
<td>Displays the shopping cart to check out.</td>
</tr>
</tbody>
</table>

The mobile app UI displays an input type appropriate to the catalog variable type. For example, for a choice variable the mobile app UI displays a list of choices.

![Sample catalog variable input](image)

**Figure 383: Sample catalog variable input**

The mobile interface for selecting and entering variables consists of the following elements.
Table 306: Elements of the mobile UI: variable input

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current variable</td>
<td>Highlights the currently selected variable.</td>
</tr>
<tr>
<td>Previous or next variable</td>
<td>Navigate to the previous or next variable input available for the catalog item.</td>
</tr>
<tr>
<td>Variable choices</td>
<td>Select the choice to use for the catalog item.</td>
</tr>
<tr>
<td></td>
<td>Swipe to see more choices.</td>
</tr>
<tr>
<td>Done button</td>
<td>Close the list of variable choices.</td>
</tr>
</tbody>
</table>

Mobile shopping cart
The mobile app UI displays the shopping cart as a separate page overlaying the service catalog.
Figure 384: Elements of the mobile UI: shopping cart

The mobile app UI shopping cart consists of the following elements.

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel button</td>
<td>Closes the shopping cart.</td>
</tr>
<tr>
<td>Items requested</td>
<td>Displays the list of items in the shopping cart.</td>
</tr>
<tr>
<td>Item costs</td>
<td>Displays the one-time and ongoing costs of each requested item.</td>
</tr>
<tr>
<td>Cart totals</td>
<td>Displays the one-time and ongoing total costs.</td>
</tr>
<tr>
<td>Checkout button</td>
<td>Requests the catalog items in the shopping cart.</td>
</tr>
</tbody>
</table>
When editing the shopping cart the mobile app interface consists of these elements.

### Table 308: Elements of the mobile UI: editing the shopping cart

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items requested</td>
<td>Select items to remove from the shopping cart.</td>
</tr>
<tr>
<td>Item quantity</td>
<td>Edit the number of items requested.</td>
</tr>
<tr>
<td>UI element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cart totals</td>
<td>Displays the one-time and ongoing total costs as you edit the shopping cart.</td>
</tr>
<tr>
<td>Save button</td>
<td>Update the shopping cart with the current items and quantities.</td>
</tr>
</tbody>
</table>

Figure 386: Elements of the mobile UI: order status

After checking out, the mobile app UI displays an order status page consisting of the following elements.
Table 309: Elements of the mobile UI: order status

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items requested</td>
<td>Displays the items the user requested from the service catalog.</td>
</tr>
<tr>
<td>Item costs</td>
<td>Displays the one-time and ongoing costs of the items requested from the service catalog.</td>
</tr>
<tr>
<td>Order details</td>
<td>Displays information from the request record for this order.</td>
</tr>
</tbody>
</table>

Mobile Connect Chat

The mobile app UI provides a native mobile interface for Connect Chat.

The Connect plugin must be activated to use Connect Chat in the mobile app UI. Many, but not all of the features supported in the desktop interface are available in the mobile app UI.
Figure 387: Elements of the mobile UI: Connect Chat
The mobile Connect Chat user interface consists of the following elements.

**Table 310: Elements of the mobile UI: Connect Chat**

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back button</td>
<td>Returns to the previous screen.</td>
</tr>
<tr>
<td>Edit button</td>
<td>Turns on conversation editing mode, in which you can delete conversations or mark them as read. Deleting a conversation only removes it from the list. The history is preserved and you can add the conversation back to your list using the new conversation button.</td>
</tr>
<tr>
<td>New conversation</td>
<td>Opens a new conversation to which you add one or more members.</td>
</tr>
<tr>
<td>Filter</td>
<td>Enables you to filter conversations by member name.</td>
</tr>
<tr>
<td>Conversations</td>
<td>Lists your open conversations. You can view All your conversations or only conversations with Unread messages.</td>
</tr>
<tr>
<td>Application navigator button</td>
<td>Displays the list of menus and modules.</td>
</tr>
</tbody>
</table>

**Mobile Connect Chat conversations**

The mobile app UI provides a native mobile interface for messaging in Connect Chat conversations.
Have you seen this?

ITIL User 6 min

System Administrator has been added to the group

System Administrator 5 min

that’s not your car is it??

ITIL User 5 min

No! So glad I parked on the street today!

incident.do?
sys_id=23d487264f8c12002fa02f1e0210c746&sysparm_record_target=incident&sysparm_record_row=2&sysparm_record_rows=32&sysparm_record_list=active

Parking garage flooded!

INC0010004
Mobile Connect Chat conversations consist of the following elements.

### Table 311: Elements of the mobile UI: Connect Chat conversation

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conversation name</td>
<td>Displays the name of the conversation.</td>
</tr>
</tbody>
</table>
| Conversation details| Opens the conversation details page, which displays the following information.  
|                     | • Record details (record conversations only)  
|                     | • Push notification preferences  
|                     | • Conversation members  
|                     | For group and record conversations, the conversation details page also provides capabilities to add or remove conversation members and to leave the conversation. |
| Avatar              | Displays an image or initials to represent a user. Tap an avatar to view details about the user, including email address and online presence status. |
| Attachment button   | Enables you to include attachments in a message, including photos and documents.                                                             |
| Message field       | Enables you to enter and send messages.  
|                     | **Note:** For record conversations, all messages are sent as comments. The ability to send a message as a work note is not supported. |

### Mobile device location

The mobile app UI can request a mobile device’s current location and store GPS coordinates in a string field.

Administrators can take advantage of mobile device tracking features such as GPS location by configuring a string field to use the `current_location` dictionary attribute. This attribute causes the mobile app UI to display a special icon to request the mobile device’s current location.
A current location field consists of the following elements.

**Table 312: Elements of the mobile UI: current location field**

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String field</td>
<td>Displays and stores GPS coordinates. The field can have any label you want.</td>
</tr>
<tr>
<td>Get current location</td>
<td>Allows users to access the current location of the mobile device.</td>
</tr>
</tbody>
</table>

When users first interact with a current location field, the mobile app asks for permission to access the location of the mobile device.
Figure 390: Allow access to location

Users can change the mobile app's access to their location from their device Settings.
Mobile barcode scanning

The mobile app UI can request access to a mobile device's camera to scan and store barcodes in a string field.

Administrators can take advantage of mobile device barcode scanning features by configuring a string field to use the barcode dictionary attribute. This attribute causes the mobile app UI to display a special icon to request access to the mobile device's camera.
A barcode scanning field consists of the following elements.

<table>
<thead>
<tr>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String field</td>
<td>Displays and stores the barcode’s numeric value. The field can have any label you want.</td>
</tr>
<tr>
<td>Scan barcode</td>
<td>Allows users to access the mobile device's camera to scan a barcode.</td>
</tr>
</tbody>
</table>

When users first interact with a barcode scanning field, the mobile app asks for permission to access the mobile device's camera.
Figure 393: Allow access to camera

Users can change the mobile app's access to their camera from their device Settings.
Mobile web UI

The system uses the mobile web UI when users access the user interface from a mobile browser. The system uses your mobile device type to determine which mobile web UI to display.

- Smartphone interface
- Tablet interface

Smartphone interface

The system supports a subset of the standard functionality on smartphones and similar devices, allowing mobile users to remotely access their instances to perform common tasks.
Supported devices for the mobile browser

Mobile browser interfaces support certain devices, operating systems, and browsers.

Table 314: Devices supported by the mobile browser interface

<table>
<thead>
<tr>
<th>Device</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple® iPhone®</td>
<td>All models of iPhone running iOS 7 and above. Safari is supported.</td>
</tr>
<tr>
<td>Apple® iPod®</td>
<td>All models of iPod running iOS 7 and above. Safari is supported.</td>
</tr>
<tr>
<td>Android™</td>
<td>Android version 4.0/Ice Cream Sandwich and above. Use the latest available version of Chrome to access the mobile browser interface on Android devices. Native browsers and older versions of Chrome support major interface functionality, but have some known issues.</td>
</tr>
<tr>
<td>BlackBerry®</td>
<td>All Blackberry devices running Blackberry 10 and above. Some configuration is required.</td>
</tr>
</tbody>
</table>

Note: Do not use the mobile interface on a desktop browser except for testing purposes.

Configure a BlackBerry device

BlackBerry devices require some configuration to use the smartphone interface.

1. Open the browser application.
2. Open the menu and select Options > Browser Configuration.
3. Ensure the following options are selected.
   - Support JavaScript
   - Use Background Images
   - Show Images: On WML & HTML Pages
   - Browser Identification: BlackBerry
4. Save your changes.

Depending on the size of the BlackBerry screen and personal preference, users may also want to navigate to the General Properties browser submenu and set the default font size and minimum font size to a smaller size. Smaller text displays more content, but may be difficult for some users to read.

Features of the smartphone interface

- A property called glide.ui.m_agents defines which devices are treated as mobile. By default, this list is: iPod, BlackBerry, Android, and iPhone.
- UI policies, UI actions, client scripts, and related lists are supported.
- The device home screen can be configured to display tags, favorites, and recently-accessed lists and documents.
- Online help is available in the base system. It can be modified and you can create new help screens. The help can also be localized for multiple language support.
- Journal streams allow you to sequentially view comments and work notes associated with a group of records, such as incidents.
Unsupported features

The smartphone interface supports a limited subset of functionality that enables users to perform common tasks.

The following capabilities are not currently supported in mobile browsers, though they may work to varying degrees.

- Switching to the standard browser interface from the smartphone interface
- HTML fields
- CODE tags to render HTML in Journal Fields
- Live feed
- Connect
- Chat
- Field styles
- Visual task boards
- Timeline visualizations
- Assessments, surveys, or legacy surveys
- Embedded lists
- List UI actions
- Mobile service catalog features
  - Order guides
  - Wizards
  - Content items
- These variable types:
  - Macro With Label
  - UI Page
  - List Collector
  - HTML
  - Macro
  - Label
  - Break
- Data lookup rules
- Advanced reference qualifiers
- Script include calls in reference qualifiers
- Custom auto-complete scripts

Smartphone interface use
The smartphone interface supports many of the features found in the standard browser.

Smartphone access to an instance
There are two methods for accessing an instance on a mobile device.

You can access the smartphone interface from a supported mobile browser or from the mobile app.

Access an instance from a mobile browser
You can access an instance using a supported mobile browser.

On a supported smartphone, navigate to the URL of an instance in a supported mobile browser. The instance automatically detects the device and redirects to the smartphone interface by appending $m.do to the end of the URL. Accessing the smartphone interface does not prevent the user from accessing the tablet or desktop interfaces.
Add a shortcut to the home screen
You can add an Apple home screen shortcut that opens a smartphone interface page.

1. In Safari, navigate to a page that you want to access as a shortcut, such as a list of open incidents.
2. Tap the sharing icon ( ) at the bottom of the browser.
3. Tap the ServiceNow Add to Home Screen icon.
4. Enter a descriptive name for the page and tap Add. The shortcut is saved to your device.

Change the home screen icon
Administrators can change the icon that appears for Apple home screen shortcuts to the smartphone interface.

To replace the default icon:
1. In the desktop interface, navigate to Self-Service > My Profile.
2. Switch to the Default view.
3. Click the reference icon by the Company field to open the company record.
4. Configure the form to add the Apple icon field.
5. Upload an image up to 57x57 pixels in the Apple icon field.

When users associated with the company add a home screen shortcut, the shortcut uses the new icon.
Customize the home page

You can customize the smartphone interface home page to display helpful information.

The smartphone interface home page displays the following types of information by default:

- Favorites
- Recent Lists
- Tags
- Recent Documents

![Figure 398: Home page](image)

To change the types of information on the home page:

1. Tap the gear icon (⚙).
   The tools screen opens.
2. Tap **Show section** to enable or disable a section on the home page.
3. Customize the number of **Items to show** for each section.
4. Tap **Horizontal scroll** to enable or disable horizontal scrolling for each section, where applicable.

Add a home page favorite
You can add any list, record, or module to the **Favorites** section of the smartphone interface home page.

The items you add to your favorites appear as tags on the smartphone interface home page. Use the favorites for quick access to the items that you use most often.

To add an item to the **Favorites** section:
1. Navigate to the screen that you want to add as a favorite.
2. Tap and hold the home icon (Home).
3. Give the favorite a name, assign its tag a color and icon, and tap **Save**.
Edit a home page favorite

Change the appearance of favorites in the app home page.

1. On the app home page, long press a favorite icon.
2. From the icon customization screen, change the icon name, color, or icon.
3. To delete a favorite, long press the icon then on iOS tap **Delete** or on Android tap the delete icon.

View or modify a mobile list

Modify a list on a mobile interface.

When you create a favorite list, it automatically appears on the mobile homepage. To see the details of a list or selection, tap its icon on the home page.

To modify a list:

1. Tap the list you want to modify.
2. Tap the filter.
3. To add an "And" rule, tap **Add an All Rule**. On an Android device, tap the add button and tap 'All' Rule.
   a) In the Add All section, click **Add New Condition**.
   b) Select a condition from the list.
   c) From the Select Operator list, select a condition qualifier.
   d) Click **Save**.

4. To add an "Or" rule, tap **Add an Any Rule**. On an Android device, tap the add button and tap 'Any' Rule.
   a) In the Add Any Rule section, tap **Add New Condition**.
b) Select a condition from the list.
c) From the Select Operator list, select a condition qualifier.
d) Click **Save**.

Both sets of conditions appear in the condition filter.

5. To delete a condition, tap the adjacent **x**.
6. Tap **Save**.

View a recently accessed record
   You can quickly access recently viewed records from the smartphone interface home page.
The **Recent Documents** section of the smartphone interface home page provides an at-a-glance summary of recently accessed records.

1. Navigate to the home page.
2. In the **Recent Documents** section, tap a record.
   The mobile view of the form appears. If the record contains attachments, they appear at the bottom of the form.
3. Tap an attachment name to open the attachment.
   If you are using the mobile app, the attachment opens in Safari.

Create a new tag
To help group and organize records, you can create tags and add them to specific records.
For example, you could create a tag called High Priority ABC and assign it to critical incidents submitted by an important customer. Tags can be defined with an icon and color for easy identification, and tags that you create can be viewed by other users in your organization. With a single tap, you or another user can then locate all critical incidents logged by ABC Corporation. You can add as many tags as needed to any record.

1. On the home page, tap the **Tag** section heading. You can also tap **My Tags** in the application navigator.
   Any previously defined tags are displayed. The image shows the number 1 in the **High Priority** tag, indicating that one record has been assigned this tag.

2. Tap the **Create or search tags** field and enter the new name. As you type, the new tag appears below the field.

3. Tap **Done**.
4. Select a color and icon for the new tag and tap **Save**.
Assign a tag to a record

To help group and organize records, you can create tags and add them to specific records.

You can add as many tags as needed to any record.

1. Navigate to a record that you want to assign a tag to, such as an incident record.
2. Tap the More icon in the lower right-hand corner of the screen and select Tags.

3. In the Tags screen, tap the tag you want to assign to this record. A check mark appears on the selected tag.

4. If needed, you can click the tag again to deselect it. You can also select multiple tags for the record.

5. Navigate away from the screen. Your tag selections are automatically saved.

**Note:** You may see green and red numbers on tags, as shown below. A green number indicates the number of records associated with that tag. A red number indicates the number of records associated with that tag that you have not viewed (that is, they were created or updated by other users).
Access tagged records

The Tags section on the smartphone interface home page shows the tags you have defined. Each tag shows the number of records assigned to it.

1. Tap the tag to display records that have been assigned that tag. The screen shows the records with that tag. Note that they are not necessarily the same kind of records. In the example below, the first record is an approval and the second is an incident.

2. You can sort the records by tapping the buttons at the top of the screen.

View a related list

Related lists for records are visible in the smartphone interface. Related lists show lists of records related to the current record.

To view a related list:

1. Open a record in the smartphone interface.
2. Tap the Related Lists tab.
3. Tap a related list.

View a journal stream

A *journal stream* is a real-time, sequential display of journal field entries, such as comments and work notes, from a group of records. You can view journal streams in the smartphone interface.

You can, for example, view a stream of incident work notes. The stream automatically aggregates all of the journal field entries from incidents and displays them, starting with the most recently updated record.

1. Navigate to a list of records that contain journal fields.
2. Tap the *Stream* icon. If no journal entries have been created for the type of records you are viewing, the *Stream* icon is not displayed.
Comments and work notes for the incidents are shown, starting with the most recent record.
Geneva    ServiceNow    ServiceNow Platform

Tutorials to use the smartphone interface
Smartphone interface tutorials.

The following video demonstrates how to navigate the smartphone interface.

Navigating the Smartphone Interface
The following video demonstrates how to customize the smartphone interface.

Customizing the Smartphone Interface

Legacy mobile UI configuration
The legacy mobile UI is available only on instances that do not use the smartphone interface.

Administrators can define which modules are available to mobile users at the application menu or module level. In the mobile UI, modules appear in sections grouped by application menu, separated by horizontal lines.

• Modules with a device type of Mobile or Any are available on mobile devices, regardless of the application menu setting.

To view all modules that are available on mobile devices, use the standard browser UI and navigate to System Definition > Modules (Mobile).

• Modules without a specified device type are available on mobile devices when the application menu setting for the default device type is Mobile or Any.

To view all application menus where modules without a device type are available on mobile devices, use the standard browser UI and navigate to System Definition > Applications (Mobile).
When accessing ServiceNow from an iPhone or Android device, users can choose whether to use the standard browser UI or the mobile UI. The user's most recent selection is saved with these user preferences:

- `mobile_use_full.iphone`
- `mobile_use_full.android`

Make module available for mobile and standard browsers

Modules are made available for mobile users, as well as standard browser users.

To make a module available for mobile and standard browsers:

1. Navigate to **System Definition > Modules**.
2. Open the module record.
3. Change the **Device type** to **Any**.
4. Enter a **Mobile title**.
   Consider providing more detail in the mobile-specific title because the application menu names are not visible in the mobile UI. For example, in the browser UI, you may define a module called Open New that appears under the Incident application menu. In the mobile view, you may call this module **Open New Incident**.
5. Enter a **View name** for this module in the mobile UI.
   Consider configuring mobile views that show a limited number of columns to improve usability for users with smaller screens or limited transmission speeds. If a view is not specified on the module record, mobile lists and forms use the Mobile view.
6. Click **Update**.

Define the mobile banner
By default, the banner at the top of the mobile UI says ServiceNow - Mobile. You can modify this text by adding a system property.

1. Enter `sys_properties.list` in the navigation filter.
2. Click New.
3. Enter the following information:
   - **Name:** glide.product.description.mobile
   - **Type:** string
   - **Value:** enter the desired mobile banner text
4. Click Submit.

Add a custom icon on the Apple home screen

Users can add a link to the ServiceNow site on the home screens of their Apple iOS devices. By default, the link appears as a ServiceNow icon. Administrators can select a custom icon instead.

1. Create the desired icon.
   
   **Note:** Use a 57x57 pixel icon, which is the maximum icon size that Apple does not automatically resize.

2. Navigate to System Properties > My Company and open the company record form.
3. **Configure the form** to add the **Apple icon** field.
4. Upload the new icon.

Define which device uses the mobile UI

You can define which devices use the mobile UI.

1. Navigate to System Properties > UI Properties.
2. Locate **Use mobile user interface if one of these strings (comma-separated) appears in the browser user_agent header** (the glide.ui.mobile_agents property).
3. Enter a comma-separated list of devices that are directed to the mobile UI.
   
   **Note:** The default list is iPod,Windows CE,BlackBerry,Android,iPhone,Opera Mini,IEMobile.

4. [Optional] If the property does not provide enough granularity, use the DetermineMobileDevice script include to define which devices use the mobile UI.
   The property value is checked first. If the user agent for the device is listed in the property, the mobile UI is used. Otherwise, the script determines which UI is used.
Disable the mobile UI
You can disable the mobile UI for all devices and direct all requests to the standard browser UI.
1. Clear the glide.ui.mobile_agents property value.
2. Deactivate the DetermineMobileDevice script include or ensure that the determine function always returns false.

Testing for mobile devices in scripts
Use the gs.isMobile() method to determine whether an action was taken through the mobile UI.

**Business Rule**

To write different messages to the log for actions through the mobile UI and through the standard browser UI, use the following script in a business rule.

```java
if (gs.isMobile())
    gs.print("submitted from mobile UI");
else
    gs.print("NOT submitted from mobile UI");
```

**UI Action Conditions (not client)**

To prevent a UI action from appearing in the mobile interface, use the following condition.

`gs.isMobile()`

**Tablet support**

ServiceNow supports nearly full-product functionality on the Apple iPad and Android tablets.

You can enjoy the mobility of the tablet device to access and manage IT service management tasks through a user-friendly tablet interface. In addition, you can take advantage of the same level of performance, availability, continuity of service, and data security that you have come to expect from ServiceNow.

**Supported tablets, operating systems, and browsers**
The following tablets, operating systems, and browsers are supported.

Apple® iOS: All models of iPad® running iOS 7 and above are supported. Safari® is supported.

Android™: Any Android tablet running OS 4.0.3 and above with the latest available version of Chrome™. Native browsers and older versions of Chrome support major tablet interface functionality, but have some known issues. The tablet UI is optimized for the iPad form factor and resolution. Not all Android tablets have the same form factor and resolution. Gesture based zoom functionality is not supported.

**Note:** The tablet interface should not be used on desktop browsers except for testing purposes.

**Features with limited tablet interface support**
Some capabilities are limited in the tablet interface.

- Editing lists: You cannot edit field values in a record from the list view. Access the record form to modify any field values.
- Dependency Views map, schema map, graphical workflow editor, Gantt chart, and visual dispatch tool: Graphics-based tools can be viewed but not modified from the tablet interface. The data presented by these tools is read-only when accessed through the tablet interface.
• Calendars: You can access calendar reports but you might not be able to scroll around the calendar as you would on the desktop interface.

• Impersonating other users: Administrators must log out and log back in as another user to see that user's access rights and views. You cannot impersonate another user while staying logged in with your normal credentials as you can on the desktop interface.

• Homepage bookmarks: UI11 bookmarks are not supported in the tablet UI. You can use favorites to mark modules in the application navigator for quick access when using the tablet UI. Note that favorites on the tablet and bookmarks on the desktop are separate features and do not synchronize with each other.

• Video and image attachment upload: iOS 6 and Android allow you to attach videos and images. Other file type attachments are not allowed.

Features that are not supported in the tablet interface
Some capabilities are not supported in the tablet interface.

• Field watcher: Administrators must use the desktop version to access the Field Watcher. JavaScript debugger:

• JavaScript debugger: Administrators must use the desktop version to access the JavaScript debug window.

• Language picker: Even if the internationalization plugin is enabled, the language picker does not appear in the tablet UI toolbar. The language selected through the desktop interface applies to the tablet UI.

• Domain picker: Tablet users cannot select any other domains that administrators configure for domain-specific personalizations. To select a new domain, use the desktop interface.

• Suffix in the navigation filter: You can use the .list, .do, or .form shortcuts to access a list of records in a table or a new form from the desktop version only.

• Help desk chat: End users cannot request a chat session nor can support technicians respond to chat requests when using the tablet interface. Help desk chat is only supported in the desktop interface.

• Printer friendly view: This view, which shows the current screen in a pop-up window without frames and the application navigator, is not available from the tablet.

• Timeline sliders and the Timeline Metrics UI actions: Features that use timelines, such as the workflow timeline and the Gantt chart are not supported from the tablet.

Tablet interface
The tablet interface provides users the flexibility to manage IT service management tasks through a user-friendly tablet interface.

This article explains unique features for the tablet interface and provides instructions for navigating ServiceNow with a tablet.

Several important actions, such as using the application navigator, accessing context menus, selecting items from pick lists, and saving bookmarks, have been enhanced.

Accessing an instance
On a mobile device that meets the tablet support requirements, you can access the standard URL of an instance. The instance automatically detects the tablet and redirects to the tablet interface by appending $tablet.do# to the end of the URL. Accessing either the tablet or desktop interface does not prevent access to the other.

Using the application navigator
The application navigator displays all the application menus you would normally see in the desktop interface.

To open the application navigator, tap the Menu icon
 on the top-left side of the interface.

Figure 399: Tablet nav button menu

Scroll the navigator up or down using one or two fingers. To expand a menu, tap the menu name. You can also filter the menu using the same filter field you see on the desktop interface just above the application navigator.

Rotating the tablet
When you rotate the tablet, the navigator behavior changes slightly.
Open the navigator by tapping the iPad menu icon (
When you rotate the tablet, the navigator behavior changes slightly. When you open the navigator while holding the tablet in landscape mode, the screen resizes so that it is all visible to the right of the navigator. When you open the navigator while holding the tablet in portrait mode, the navigator overlays the left side of the screen. If the navigator is open when you rotate the tablet, the navigator automatically closes.

Figure 400: Tablet landscape

Floating the application navigator

When the tablet is in landscape mode, you can float the application navigator. This means that the application navigator floats over the left-hand side of the screen. While working full-screen, you can open the application navigator, tap the menu or module you want to view, and close the application navigator without changing the appearance of your screen.

To turn on the floating navigator, tap the gear icon ( ) and select **Use Floating Navigator**.

Managing favorites and shortcuts
The favorites menu in the application navigator shows you only the items marked as favorites.

Use the favorites menu to access the items that you use most often. On the application navigator, tap the star icon to add an item to your favorites. You can add items that pertain specifically to you, such as incidents assigned to you. After you add a favorite, the star icon changes to dark gray. To remove a favorite, tap the star icon again.

Modules you access are automatically marked as favorites, by default. You can enable or disable automatic favorites with the Automatically Add Favorites option from the main menu.

![Favorites menu](image)

**Figure 401: Favorites menu**

Access the favorites menu

Access the favorites menu from a tablet web browser.

1. Tap the favorites icon. The icon stays depressed.

2. Tap the application navigator icon. When the favorites icon is depressed, only favorites appear in the application navigator.
**Note:** Bookmarks in the desktop version are not the same as favorites in the tablet version. Bookmarks cannot be accessed or modified through the tablet interface.

Create a Safari shortcut

Use the sharing feature on the Safari browser to create a shortcut to any ServiceNow page.

The shortcut is saved on your tablet just like an application. Use shortcuts to quickly access a specific ServiceNow page with one touch.

1. Navigate to a frequently used page, such as a list of open incidents.
2. Tap the sharing icon on the Safari browser header.
3. Tap the ServiceNow **Add to Home Screen** icon.
4. Enter a descriptive name for the page and tap **Add**. The shortcut is saved to your device.
Preview pane

The tablet interface provides a handy split-screen view that preserves a list of records in one pane while displaying a read-only preview of a record in another pane.

Use this view to preview several records in a list without having to navigate back and forth between lists and forms. The preview pane is available for all lists.

To open the split-screen view, tap the preview pane icon ( ). The icon becomes green and a read-only view of the form opens on the right.
Figure 402: Preview pane

To close the preview pane, either tap the Close icon (X) or tap the green preview-pane icon.

**Note:** If the navigation pane is open when the preview-pane icon is tapped, the navigation pane will close.

Add comments or work notes from the preview pane

You can add comments directly from the preview pane to any record that has an *Additional Comments* journal field.

This allows you to enter comments or work notes without scrolling down the form. Comments and work notes are updated in real time.

1. Tap **Comment**. The **Comments and Work Notes** pane appears.
2. Enter content in the text field.
3. Tap **Comment** or **Work note** to add the content to the relevant journal field.
View a recently accessed document

The Recently Accessed Documents screen provides an at-a-glance summary of recently accessed records.

Avoid tapping through the application navigator to access a record that you need frequently or a record that you just viewed.

1. Tap the document summary icon in the banner. The most recently accessed records appear at the top. A list of important fields appears on each record summary.

2. To open a record from this view, tap the record name.
3. To open a read-only preview of the record in another pane, tap the preview pane icon. Notice that the preview pane icon turns green
4. To close the preview pane, tap the green preview pane icon again.

Accessing the list header menu
Access the list header context menu by tapping the icon in the list header.

Tap the menu icon () in the list header.
Figure 403: List header menu iPad

Sorting by column header works the same as the standard interface. Tap the column name to sort by that column.

Accessing the list row menu

How to access the list row menu.

Tap any cell to see the list row context menu with actions related to the values in that row.
Accessing the form menu

How to access the form menu.

Tap the form name or icon to access the form context menu.

Toggling between related list views

When using a tablet to view forms that include multiple sections, such as the Incident form, you can tap Toggle Tabs to toggle between tabbed and sequential arrangements of sections.
Tap the Toggle tabs icon ( ) to toggle between tabbed and sequential arrangements of sections.

Figure 405: Toggle tabs
In sequential view, you can expand or collapse a section by tapping **expand** (+) or **collapse** (-) in the section header.

Select a related record
The tablet interface provides a split-screen record picker so you can easily add and remove related records.

Related lists allow you to **relate a record** to the record currently displayed in the form.

To select related records:

1. Navigate to a record form. In the example below, an incident record is related to a problem.
2. Tap the incident number. The screen splits to display information associated with the incident. The way that the related list area appears can be changed by tapping the Toggle Tabs button.

3. Tap the Related Records tab to view a summary of the related problem.
4. You can view or edit the problem record by tapping the preview pane icon next to the problem number.
Lookup lists
To filter the lookup results, use the filter fields above each column.

You can also perform a contains search in the filter fields using a period (\(\cdot\)). For example, to search for all names that contain the characters tu, enter the following: \(\cdot\)tu.

![Lookup list iPad](image)

**Figure 407: Lookup list iPad**

Accessing another record from a form
Tap the reference icon to jump to another record.

On the tablet, tapping the reference icon ( ) opens the related record in the full frame, rather than providing the pop-up overlay that appears on the desktop interface.
Glide list items

The tablet interface provides an easy way to interact with items that use the glide_list field type, such as watch lists.

The following example illustrates how to add a name to a watch list.

1. On a form with a watch list field, tap the lookup icon.

2. Filter the watch list by entering text in the search field above any column. The list is automatically filtered as you type. You can also perform a contains search in the filter fields by using the period (.) . For example, to search for all names that contain the characters tu, enter the following: .tu.

3. Tap the record that you want to add. The record is added to the selection column on the left.
4. To add yourself, add an email address, or clear the form, use the menu at the bottom of the selection column:

5. Tap Done when finished.

Labeling records
To help group and organize records, users can add labels, which are keywords that describe the record. For example, an incident record that deals with a wireless internet outage could be labeled with the keyword wireless. Later, someone who has access to incident records can filter all incidents by the keyword wireless to see which incidents pertain to that topic. Users can add as many labels as needed to any record. Labels are not case sensitive.

Labels that you add to a record are saved in the Labels table with the following default settings:
- Active: true. The label is enabled and appears in the tablet and desktop interfaces.
- Global: false. The label is not globally visible. Labels are visible only to the user who created them.
- Navigation: false. Tablet labels do not appear in the application navigator on either the tablet or the desktop interface.

Administrators can modify these labels as they would modify labels in the system definition.

Assign a new label
How to assign a new label to a record.

1. Navigate to a record and tap inside the Add label text box.
2. Enter the key terms or groups of terms as you want them to appear. Spaces are allowed between multiple terms.
Access a record that has labels
How to access records that have labels.

1. Tap the recent documents icon to navigate to Recently Accessed Documents.

2. Tap the gear icon in the top banner. The list of labels shows the number of records that use each label.

3. Select a label to show only records with that label.

Edit and delete labels
How to edit label text.

1. Tap the recent documents icon to navigate to Recently Accessed Documents.

2. Tap the gear icon.

3. Tap Edit in the list of labels. The labels become editable.

4. Make the necessary changes or tap the X icon to delete the label.

5. Tap Save.
BlackBerry FAQ

This topic answers frequently asked questions on using ServiceNow on a BlackBerry device.

How do I configure my BlackBerry device to access ServiceNow?

What BlackBerry devices are supported?

Blackberry OS version 4.6.0 or later is required which is included on the newer BlackBerry models. To find your BlackBerry OS version, on your Blackberry, go to Settings > Options > About to find the version. Older versions cannot be supported due to the severely limited JavaScript implementation in the BlackBerry browser.

Getting response "413 Request Entity Too Large" trying to bring up ServiceNow on BlackBerry devices.

BlackBerry devices are configured to only allow files of a certain size to be downloaded. This is typically 128K. This limitation can be changed to by your corporate BlackBerry administrator.

- Can you access https://<instance name>.service-now.com with your BlackBerry without getting the "entity request too large" error? If you can, it is most likely a problem with your specific instance:
• Have your ServiceNow administrator verify that "blackberry" is still listed on the property (glide.ui.mobile_agents) which defines the list of devices to receive the mobile UI.

• If you get the "entity request too large" error when attempting to access https://<instance name>.service-now.com:
  • If your BlackBerry device is configured to masquerade as another browser, such as Internet Explorer, then the full desktop version of ServiceNow will attempt to be sent to your device and you will likely get the "request entity too large" error as the size of the main JavaScript file will probably be larger than your configured limit. Verify your device is configured to work as a BlackBerry browser as shown in step 5 of the BlackBerry Configuration article. If it incorrect, you will then need to clear your BlackBerry device's cookies after correcting the browser.
  • Have your corporate BlackBerry administrator check to see whether the maximum size has changed recently; see this article for further information: http://www.blackberryforums.com.au/forums/general-bes-discussion/284-request-entity-too-large.html

Mobile experience set up for admins

As an administrator, configure the mobile experience for your users to access an instance on a tablet or smartphone.

Requirements

Configuration for the mobile experience takes place in a regular desktop instance. Most of the configuration options are available for both the native mobile app and the mobile web experience. Use the native app or the mobile web to test your configuration.

Role required

admin

Before you begin

Take the following into consideration before configuring the mobile experience:

• What applications and modules do you want to be accessible from a mobile device?
• How do you want mobile lists to appear on a mobile device?
• Which users or roles should have access to Connect Chat?

What to do

Activate the mobile interface

If you are using the native mobile app, download the app from the iTunes store or from the Google Play store. Make sure you are on an instance using Geneva patch 6 or higher to use the Android app.
If you are accessing an instance on a mobile device or tablet from the web, **activate the mobile web interface**.

**Configure the application menu**
Determine which **applications** and **modules** you want to appear in the mobile application navigator.

**Set up mobile lists**
Use the **table titles configuration option** to determine the length of a title in a mobile list, as well as the default fields that appear in a table.
**Customize sortable columns** in a mobile list.

**Define homepage favorites**
Set up the **default homepage favorites** for your users.

**Configure Connect Chat for mobile**
Determine which users or roles need access to **Connect Chat on a mobile device**.

**Configure mobile UI properties**
- **Disable the mobile or tablet browser UI** on page 1230.
- **Define devices that use the smartphone interface** on page 1230.
- Configure **additional theming options**.

**Configure the tablet UI**
Use **tablet properties** to allow access to the table UI, determine which devices have access to the tablet UI, and perform some basic custom theming.
**Configure UI actions** to determine the functionality allowed on a tablet.
Configure tablets to **run the smartphone interface** in place of the tablet interface.

**Next steps**
Most configuration items for the mobile UI are optional. Click any of the topics below for additional configuration.

**Activate the mobile web interface**
Mobile web interfaces are automatically active on new instances. For instances upgrading from an earlier version, an administrator can activate the Mobile UI plugin (com.glide.ui.m) if it is not already active.

Role required: admin

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

**Smartphone interface application navigator**

Administrators can define which application menus and modules are available on the smartphone interface application navigator.

For example, the screen below shows the INCIDENT and USER ADMINISTRATION application menus.

![Smartphone modules](image)

**Figure 409: Smartphone modules**

Define a new smartphone application menu

Administrators can define which application menus are available on the smartphone interface application navigator.

1. In the standard browser interface, navigate to **System Mobile UI > Navigator Apps**.
2. Click **New**.

3. Fill in the fields, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the application menu.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to specify the order of the menu. For example, an entry of 100 would place this application menu before one with an <strong>Order</strong> entry of 200.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock icon to select the roles for this application menu. Only users with the designated roles can access the modules under this application menu.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate this application menu.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

**Note:** The new application menu does not appear on the smartphone interface application navigator until at least one module is added.

Define a new smartphone module
After defining the application menu, you can define modules and assign them to the new application menu.
For more information on defining the application menu, see *Define a new smartphone application menu* on page 1203.

1. On your instance, navigate to **System Mobile UI > Navigator Apps**.
2. Open the application menu you want to create the new module for.

### Modules

<table>
<thead>
<tr>
<th>Name</th>
<th>Order</th>
<th>Active</th>
<th>Table</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create New</td>
<td>10</td>
<td>true</td>
<td>Incident</td>
<td>form/incident-1</td>
</tr>
<tr>
<td>Active</td>
<td>20</td>
<td>true</td>
<td>Incident</td>
<td>[incident]</td>
</tr>
<tr>
<td>Active P1</td>
<td>30</td>
<td>true</td>
<td>Incident</td>
<td>[incident]</td>
</tr>
<tr>
<td>Assigned to Me</td>
<td>40</td>
<td>true</td>
<td>Incident</td>
<td>[incident]</td>
</tr>
<tr>
<td>Stream P1</td>
<td>60</td>
<td>true</td>
<td></td>
<td>streams/incident&amp;p=q:active=true*priority=1</td>
</tr>
<tr>
<td>Filter</td>
<td>100</td>
<td>true</td>
<td>Incident</td>
<td>list_filter/incident/create</td>
</tr>
</tbody>
</table>

3. Under **Modules**, click **New**.
4. Fill in the fields, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the module.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to specify the order of the module within the application menu. For example, an entry of 100 would place this module before one with an <strong>Order</strong> entry of 200.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application menu</td>
<td>Displays the application menu from which you accessed this screen. Select a different application menu, if appropriate.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the table for this module.</td>
</tr>
<tr>
<td>Updated</td>
<td>Displays the date and time when the module record is updated.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock icon and select the roles for this module. Only users with the designated roles can access this module.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate this module. Only active modules appear in the application menu.</td>
</tr>
<tr>
<td>Filter</td>
<td>Create a filter for identifying which fields this module uses from the selected Table.</td>
</tr>
<tr>
<td>Path</td>
<td>[Optional] Enter a custom URL for the module. The URL must be in this format: type_of_link/table/parameters.</td>
</tr>
</tbody>
</table>

**Note:** If you are creating a module for a map page, see the procedure specific to map pages.

5. Click **Submit**.
Configure the mobile list layout
Configure the way forms display in the mobile UI.

1. From an instance on your desktop, navigate to any form you want to configure, for example, Incidents.
2. Click the list header menu and navigate to Configure > List Layout.
3. From the View name list, select **Mobile**.
4. Use the slushbucket to add or remove fields from the mobile list.
Configuring Incidents List

Available
- Created
- Created by
- Date format
- Default perspective [+]
- Department [+]
- Domain [+]
- Domain Path
- Email
- Employee number
- Failed login attempts
- Field3 [+]
- Gender
- Geolocation tracked
- Home phone
- Internal Integration User
- LDAP server [+]
- Language
- Last login

List view
- View name: Mobile

Related Links
Show versions
Any fields you add to the list appear as a sortable column and as a field in the card of a specific record in the mobile UI.

*Configure mobile list search fields*

Configure search fields to determine what fields are searched for in the table. If search fields aren't configured, only the columns in the mobile list layout are searched.

Search fields might not appear on the Table Title form. Add **Search Fields** to the form by editing the form layout.

1. Navigate to **System Mobile UI > Table Titles**.
2. Open the table you want to configure search fields for, or click **New**.
3. Use the slushbucket to add or remove fields from the mobile list.
Define a new smartphone table title
Administrators can define unique titles for all tables viewed in the smartphone interface. This can be useful for creating shorter display values to accommodate the smaller screen size of the smartphone.

1. In the desktop interface, navigate to **System Mobile UI > Table Titles**.

2. Click **New**.

3. Fill in the fields, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select the table you are defining a title for.</td>
</tr>
</tbody>
</table>

**Note**: The list shows only tables and database views that are in the same scope as the title.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields</td>
<td>Click the lock icon to display the Fields slushbucket. Use the slushbucket to select the fields that display by default when this table is viewed in the smartphone interface.</td>
</tr>
<tr>
<td>Updated</td>
<td>Displays the date and time when the table title was updated.</td>
</tr>
<tr>
<td>Script</td>
<td>[Optional] Create a server-side script that generates a string to display under the display value on lists. For example, if you are defining a title for My Approvals, the script could display the different approval items below the title.</td>
</tr>
</tbody>
</table>

4. Click Submit.

**Define default smartphone home page favorites**
You can configure the default items for the Favorites area of the mobile home page, including the colors and icons used.

1. On the device, add a homepage favorite for a particular module. For instructions, see Add a home page favorite on page 1166.
   
   For example, you could create a favorite for My Incidents.

   ![Favorites]

   2. In the desktop interface, navigate to System Mobile UI > Home Page Favorites.
      
      Notice that My Incidents is displayed.

   3. Click My Incidents.
4. To make the **My Incidents** tag a default for all mobile users, clear the **User** field.
5. Click **Update**.

**Remove smartphone default home page favorites**

To remove default favorites, remove them from the **Home Favorites** list in the desktop interface.

1. In the desktop interface, navigate to **System Mobile UI > Home Page Favorites**.
2. Locate the favorite you want to remove.
3. Select the adjacent check box.
4. In the actions choice list, select **Delete**.
5. Click **OK**.

**Configure Connect Chat for mobile**

Configure Connect Chat to show or hide on the mobile app homepage.

Connect Chat appears on the mobile app homepage by default if it is active on the instance and the user has the correct roles. As an administrator, you can hide Connect Chat by deactivating the plugin, or by configuring the roles for certain users. Use these steps to show Connect Chat for users with a particular role.

1. Navigate to **Collaborate > Administration > Properties**. Alternatively, in System Properties, search for **connect.role**.
2. In the **Comma separated whitelist of roles able access Connect. Empty allows all roles.**, type a list of the roles you want to enable Connect Chat for.

**Configure the blur app option**

As a security feature, administrators can configure the mobile app to blur when not in focus on a mobile device.
For example, when you double-click the home button on your mobile device to close apps or navigate back to where you left off, the ServiceNow app appears blurred.

1. Navigate to **System Properties > Mobile UI Properties**.
2. Select the **Blur native app UI when the application enters the background** check box.

![Figure 410: Blurred app](image)

**Online smartphone help**

The smartphone interface includes several help screens that can be used as is or modified to suit your users’ needs.
You can also create new help screens. They can also be localized to support multiple languages.

**Note:** A basic knowledge of wiki markup is needed for editing the content.

Edit the existing smartphone help text

Users with the admin role can edit the existing smartphone help text.

The default help screens are organized based on the *index* file. That file contains links to the other help files and presents those links on the device as a menu. If items are added to or removed from this file, the resulting menu changes accordingly.

1. Navigate to **System Mobile UI > Mobile Help**.

   ![System Help Table](image)

   **Figure 411: System help**

   2. Click the **Updated** link of the help file you want to edit.
3. Click **WikiText**.
4. Edit the text and wiki markup as needed.

![Wiki text example]

**Figure 413: Sample wiki text**

5. Click **Update**.

Create a new smartphone help screen

Users with the admin role can create a new help screen.

1. Navigate to **System Mobile UI > Mobile Help**.
2. Click **New**.
3. In the *Path* field, enter a name for the new help screen; for example, *adding_aShortcut*.
   
   Spaces are not allowed in the name.
4. Click WikiText.

5. In the editing box, enter the text for the new help screen, using wiki markup to define the formatting.
Adding a Shortcut

# Navigate to a page that you want to access as a shortcut, such as a list of open incidents.
# Tap the sharing icon at the bottom of the browser.
# Tap the ServiceNow "Add to Home Screen" icon.
# Type a name for the page and tap "Add".

:::[[image:Shortcut.png|300px|left]]<br style="clear:both;" />
6. Click **Submit**.
7. On the list screen, open the file called **index**.
8. Click **Wikitext**.

9. Insert a line where you want the new help screen to appear and specify the name of the new page.

10. Click **Update**.
    
    The device shows the new help menu item and help content.
Display localized smartphone help content

You can edit and translate existing help screens so that help content in multiple languages is available.

Role required: admin
After help files have been translated, the smartphone interface uses the help screens identified by the [user-specific language] in the User [sys_user] table.

1. Navigate to System Mobile UI > Mobile Help.
2. Open the help file you want to translate.
3. Translate the existing help text. You can also insert translated images, as needed.
4. In the Language field, select the language of the translated text.
5. Click Update.
Enable UI policies for the smartphone interface
UI policies can be defined to run on forms on the smartphone interface, the desktop UI, or both.

1. In the desktop interface, navigate to System UI > UI Policies.
2. Create a UI policy that is compatible with the smartphone interface.
3. **Configure** the UI Policies form to add the **Run scripts in UI type** field.
4. Select the **Run scripts** check box.
5. In the **Run scripts in UI type** field, select **Mobile** or **Both** to have the policy run on the smartphone interface.

6. Complete the fields, as appropriate.
7. Click **Submit**.
Enable client scripts for the smartphone interface

Client scripts can be defined to run on forms on the smartphone interface, the desktop UI, or both.

1. In the desktop interface, navigate to System Definition > Client Scripts.
2. Create a client script that is compatible with the smartphone interface. To ensure this, see Client script types on page 3901.
3. In the UI Type field, select Mobile or Both to have the script run on the smartphone interface.

   **Note:** You may need to configure the form to add the UI Type field.

4. Fill in the fields, as appropriate.
5. Click Submit.
Enable mobile location and barcode scanning

Take advantage of mobile devices by allowing location and barcode scanning in the mobile app.

Role required: admin

1. Navigate to the form you want to add the location or barcode scanner to.
2. Use the form context menu to switch the form view to Mobile.
3. From the form context menu, navigate to **Configure > Form Layout**.
4. In the View name list, make sure **Mobile** is selected.
5. Use the Create new field section to add a location or barcode field to the form. Set the field type to **String**.
6. From the form, right-click the new field and click **Configure Dictionary**.
7. In the Attributes related list, click **New**.
8. Use the Attribute field on the Dictionary Attribute form to search for **barcode** or **location**.

The added fields appear on the form in the native mobile app. This feature is not available on the mobile web. Users will still need to allow the app to access their location and camera on their devices. These fields only appear when you create a new record, they won't appear in one that is already existing.

**Smartphone interface UI actions**

UI actions function the same on the smartphone interface as on desktop interface, but are keyed off a different table: UI Action - Mobile [sys_ui_ng_action]. You can add these kinds of UI actions in the smartphone interface:

- list buttons
- form buttons
- form more items (these are items that display when the user taps the **More** button)

![UI Actions - Mobile](image)

**Figure 414: UI action**

The lowest number in the **Order** column identifies the primary button at the top of the form.

**Redirect to a URL in the smartphone interface**

As part of a UI action, you may want to redirect a user to a URL in the smartphone interface. For example, you might add links to a form or open a new record after it is created from a UI action.
• Use this syntax to define the redirect link:

```javascript
action.setRedirectURL('incident.do?sys_id=-1');
```

**Note:** This functionality works the same as in the desktop version. For more information, click here.

---

Back navigation in the mobile interface

Mobile UI actions support back navigation when a UI action finishes. You can navigate back one logical navigation item in the current navigation stack.

When the `navigate_back` flag is set to true for a mobile UI action in the UI Action - Mobile [sys_ui_ng_action] table, you can move back to the previous screen.

Sometimes one or more discrete screens, or a nested or partial view of the screen, can represent a single navigation item. For this reason, back navigation can result in different UI behaviors depending on the current navigation content or device.

For example, the activity stream and form for a record are considered as one logical navigation item (the current record). However, some devices use multiple discrete screens for this navigation. When back navigation is triggered from a UI action on a form, multiple screens may be dismissed to return to a previous item, such as a list.

**Note:** Setting a redirect on the UI action through `action.setRedirectURL();` takes precedence over the `navigate_back` flag.

---

Define devices that use the smartphone interface

Users with the admin role can define the devices that use the smartphone interface.

1. Navigate to System Properties > UI Properties.
2. Locate *Use new mobile user interface if one of these strings (comma-separated) appears in the browser user_agent header* (the `glide.ui.m_agents` property).
3. Enter the types of devices that are directed to the smartphone interface.
   
   The default list is iPod, BlackBerry, Android, and iPhone.

Disable the mobile or tablet browser UI

Administrators can disable the mobile web interface for all devices and direct all activity to the standard browser UI.

This task applies specifically to users accessing an instance from a mobile browser, not the mobile app. System properties check the User Agent string of a browser for keywords found in that property’s value. If the User Agent string finds a corresponding property value, it directs to that indicated UI. For example, a property like `iPhone` would direct to a mobile UI.

**Table 315: UI Agent Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>glide.ui.m_agents</code></td>
<td>Controls which browser is directed to the mobile UI</td>
<td>iPhone, android_phone, IE Mobile, Windows Phone, iPod, Windows CE, BlackBerry, BB10</td>
</tr>
<tr>
<td><code>glide.ui.mobile_agents</code></td>
<td>Use for the legacy mobile UI. It serves the same purpose as <code>glide.ui.m_agents</code>, but has no effect on access to the current mobile UI</td>
<td>iPod, Windows CE, BlackBerry, Android, Opera Mini, IE Mobile, Windows phone, iPhone</td>
</tr>
</tbody>
</table>
To change the rules for redirection:

- Navigate to **System Properties > Mobile UI properties** and clear the glide.ui.m_agents property value, which contains the list of mobile devices. Clearing the values in this field does not prevent access to the instance on mobile devices, but directs mobile device users to the desktop UI.

**Note:** Modifying these properties only changes the rules for redirection. Anyone can still access the mobile UI using $m.do.

- To eliminate access to the mobile or tablet UI completely, create a UI page to override $m or $tablet and redirect to a new page.

### Administering the tablet UI

Administrators can define whether the tablet UI is available and certain aspects of its appearance with a series of system properties.

Administrators can define whether the tablet UI is available and certain aspects of its appearance with a series of system properties. Additionally, administrators can configure access to UI actions based on whether the tablet UI is used.

### Managing tablet properties

Administrators can modify the following properties by navigating to **System Properties > Tablet UI Properties** or the System Properties [sys_properties] table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable tablet UI. (glide.ui.tablet_enabled)</td>
<td>Select <strong>Yes</strong> to allow users to access a ServiceNow instance from a supported tablet. Clear the check box to disable the tablet UI. This option is enabled by default.</td>
</tr>
<tr>
<td>Use tablet user interface if one of these strings. (comma-separated) appears in the browser user_agent header. (glide.ui.tablet_agents)</td>
<td>Enter the text strings that the ServiceNow instance looks for in the user agent of the tablet. If any of the strings in the property are found, the tablet interface is used. The default value is <em>ipad</em>.</td>
</tr>
<tr>
<td>Brief page title for tablet interface, also used for home screen icon labels. The iOS will truncate the title to 13 characters when adding it to the home screen. (glide.ui.tablet.title)</td>
<td>Enter the text that appears on the browser tab to indicate which website is currently loaded in the browser. The default value is <strong>ServiceNow</strong>.</td>
</tr>
<tr>
<td>Default home screen icon label used in iOS6+. (glide.ui.tablet.title.ios_webapp)</td>
<td>Enter the text that appears when you use the Safari share feature to save a ServiceNow page as an icon label on the tablet home screen. The default value is ServiceNow.</td>
</tr>
<tr>
<td>Appears on the right-hand side of the tablet interface header. (tablet.header.text)</td>
<td>Enter the text that appears on the right side of the tablet header. The default value is <strong>servicenow</strong>.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Appears in the tablet footer. (tablet.footer.text)</td>
<td>Enter the text that appears at the bottom of the tablet screen. The default value is Copyright ServiceNow 2013.</td>
</tr>
<tr>
<td>Start color of gradient for the tablet UI header. (css.tablet.gradient.start)</td>
<td>Enter the hexadecimal value for the color at the top of the gradient on the UI header. The default value is #666666 (gray).</td>
</tr>
<tr>
<td>End color of gradient for the tablet UI header. (css.tablet.gradient.end)</td>
<td>Enter the hexadecimal value for the color at the bottom of the gradient on the UI header. The default value is #111111 (black).</td>
</tr>
<tr>
<td>Color for text and icons in the tablet header and footer. (css.tablet.headerfooter.text.color)</td>
<td>Enter the hexadecimal value for the color of the text and icons on the tablet interface. The default color is #d3d3d3 or lightgrey.</td>
</tr>
</tbody>
</table>

Configuring UI actions and view rules for the tablet interface
Administrators can configure access to UI actions based on whether the tablet interface is used.

You could, for example, reduce the number of UI actions that appear on the tablet interface, limiting the UI to the most basic functions that tablet users need.

To determine if a user is accessing an instance from a tablet, create a script using the GlideMobileExtensions.runningTablet() method. To create a view rule that applies to tablet users only:

1. Navigate to System UI > View Rules.
2. Click New.
3. In the Device type field, select Tablet.
4. Fill in the other fields to define the view rule.

Use the Smartphone Interface on a tablet
Administrators can configure tablets to run the smartphone interface in place of the tablet interface.

Role required: admin
By default, the system provides tablets with their own mobile web UI. If you want a tablet to run the smartphone interface instead of the tablet interface, you must make the following changes.

1. Remove the device from the list of tablet mobile agents in the glide.ui.tablets_agents property. For example, remove the ipad value.
2. Add the device to the list of smartphone mobile agents in the glide.ui.m_agents property. For example, add the ipad value.

Core configuration

Core configuration encompasses changes made to the platform as well as supporting applications. These changes can affect global settings as well as settings for particular applications.

These settings include:
- **Forms and lists**: change the behavior and appearance of forms and lists.
- **Navigation and UI and search**: configure how users browse and search in the ServiceNow platform.
- **Time and performance**: keep your organization running and on time.
- **Localization and mobile device support**: take the ServiceNow platform anywhere.
- **Plugins and out-of-box appendix**: expand the ServiceNow platform to fit your needs.

ServiceNow plugins

Plugins provide functionality within a ServiceNow instance.

Many plugins are active in the base ServiceNow system. Optional functionality is provided with plugins that administrators can activate, request, or purchase.

---

**Note:** After a plugin is active, you cannot disable or deactivate it. You can hide the functionality, if needed.

---

Activate a plugin

Follow these steps to activate the plugin.

Role required: admin

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Request a plugin

Some plugins must be activated by ServiceNow personnel. These plugins do not appear in the **System Definition > Plugins** list.

Role required: none
Request the plugin through the HI Service Portal.

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
<tr>
<td>Specify the date and time you would like this plugin to be enabled</td>
<td>Date and time must be at least 2 business days from the current time.</td>
</tr>
<tr>
<td>Reason/Comments</td>
<td>Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

**Purchase a plugin**

Some features require a separate subscription from the rest of the ServiceNow platform.

Role required: admin

To purchase a subscription, contact your ServiceNow account manager. In most cases, the account manager will arrange to have the plugin activated on your organization's production and sub-production instances, generally within a few days. In some cases, you can activate the plugin within the instance.

If you do not have an account manager, decide to delay activation after purchase, or want to evaluate the feature on a sub-production instance before purchase, follow the steps to activate a plugin. If the plugin is not listed in the **System Definition > Plugins** module, make a request through HI.

**List of Geneva plugins**

This table lists plugins that are currently available in Geneva.

<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity formatter</td>
<td>com.glide.ui_activity_formatter</td>
<td>Quickly and easily filter the list of activities, or history, on a task form.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Aggregate Web Service</td>
<td>com.glide.web_service_aggregate</td>
<td>Provides SOAP Access to GlideAggregate functionality.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Amazon Web Services</td>
<td>com.snc.aws</td>
<td>Integration with Amazon Web Services - CloudFormation, EC2</td>
<td>inactive</td>
<td></td>
<td>Orchestration - Amazon EC2, Amazon Web Services Common, Amazon Web Services Activities, Discovery Components for Amazon Web Services</td>
</tr>
<tr>
<td>Amazon Web Services Demo Data</td>
<td>com.snc.aws.demo</td>
<td>Demo data for AWS</td>
<td>inactive</td>
<td>true</td>
<td>Amazon Web Services</td>
</tr>
<tr>
<td>Angular AMB Services New in Geneva</td>
<td>com.glide.ui.ng.amb</td>
<td>Angular Services for AMB.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.ng, com.glide.amb, com.glide.record_watcher</td>
</tr>
<tr>
<td>Application Creator</td>
<td>com.snc.apps_creator</td>
<td>Retired. Replaced by the Platform as a Service plugin.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.angular, com.glide.ui.heisenberg</td>
</tr>
<tr>
<td>Application File</td>
<td>com.snc.apps_file</td>
<td>Associates configuration records with an application and tracks record metadata.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application Metadata</td>
<td>com.snc.metadata</td>
<td></td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Applications Access Control</td>
<td>com.snc.apps_access</td>
<td>Provides file-level access for application development.</td>
<td>active</td>
<td>false</td>
<td>com.snc.apps, com.snc.apps_picker</td>
</tr>
<tr>
<td>Applications Picker</td>
<td>com.snc.apps_picker</td>
<td>Allows users to select the desired application during application development.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Approvals with e-Signature</td>
<td>com.glide.e_signature_approvals</td>
<td>Adds a prompt for credentials when an approver attempts to approve a request via the list context menu or Approve UI Action on the Approval form.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Architecture Compliance</td>
<td>com.snc.architecture_compliance</td>
<td>Manages scheduled or on-demand audits of CMDB records, to determine if configuration items (CI) match expected attributes.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.certification_core</td>
</tr>
<tr>
<td>Assessment</td>
<td>com.snc.assessment</td>
<td>Provides capabilities to use custom questionnaires and scripted queries to evaluate, score, and compare any records in ServiceNow.</td>
<td>active</td>
<td>true</td>
<td>com.glideapp.workflow, com.glideapp.survey, com.snc.bestpractice.task_survey, com.glide.survey_designer</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Assessment Components</td>
<td>com.snc.assessment.application</td>
<td>Provides the core components required for legacy surveys.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.bestpractice.task_survey, com.glideapp.survey</td>
</tr>
<tr>
<td>Assessment Designer Common</td>
<td>com.glide.assessment.designer.common</td>
<td></td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.ng.dc</td>
</tr>
<tr>
<td>Asset Management</td>
<td>com.snc.asset_management</td>
<td>Provides the ability to manage all your assets, consumables, and software licenses.</td>
<td>active</td>
<td>true</td>
<td>com.snc.expense_line, com.snc.model, com.snc.organization_management, com.snc.fixed_asset, com.snc.depreciation, com.snc.automation, com.glideapp.home</td>
</tr>
<tr>
<td>Auto Recovery New in Geneva</td>
<td>com.glide.autorecovery</td>
<td>Enables Auto Recovery functionality.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Automatic Assignment New in Geneva</td>
<td>com.snc.automation</td>
<td>A task assignment application that uses tables that are children of Task [task] to use auto-assignment to automatically find eligible assignees for any task.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Automation Applications Core Available by request</td>
<td>com.snc.automation_apps_core.application</td>
<td>Provides core functionality for automation applications.</td>
<td>inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Export Set Functionality</td>
<td>com.glide.system_export_set</td>
<td></td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Best Practice - Bulk CI Changes</td>
<td>com.snc.bestpractice.bulkchange.feature_set</td>
<td>Functionality to propose affected CI changes in the change_request record that will automatically be deployed to all related task_ci records.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Best Practice - Change Risk Calculator</td>
<td>com.snc.bestpractice.change_risk.feature_set</td>
<td>Provides risk and impact calculations for change management.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Best Practice - Incident Resolution Workflow</td>
<td>com.snc.bestpractice.incident.feature_set</td>
<td>Best practices for incident resolution dictate that, rather than closing the incident, the incident should have a state of Resolved. This state gives the service desk a mechanism to verify that the caller is satisfied with the resolution, and that the customer agrees with closing the incident.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Best Practice - ITIL KPI Reports</td>
<td>com.snc.bestpractice.itil_kpi.feature_set</td>
<td>Provides a series of reports that track the Key Performance Indicators (KPI) of incident management and problem management.</td>
<td>active</td>
<td>true</td>
<td>com.glide.metrics</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Best Practice - Task Survey Management | com.snc.bestpractice.task_survey.feature_set | **Active:** true  
**Has Demo Data?:** true  
**Dependency:**  
- com.glideapp.survey  
- com.glide.db_view  
- com.glide.notification  

Enhanced survey functionality, includes triggering survey requests based on task conditions and linking survey responses to the task that generated the survey. This allows for detailed survey reporting leveraging data from the related task record. The demo data relies on incident and user demo data, but will not be installed with this plugin.  

Note: A significantly improved and updated feature named Survey Management is available starting with the Eureka release. |
| Big Instance Available by request | com.snc.big_instance | **Active:** inactive  
**Has Demo Data?:** false  

Big instance scalability technologies. |
| Boot profile support routines Available by request | com.snc.boot_profile | **Active:** inactive  
**Has Demo Data?:** false  

Common routines for installation profiles. |
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Rule V2</td>
<td>com.glide.business_rule_v2</td>
<td>business rules to support script-free conditions and behaviors.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Catalog Designer Common</td>
<td>com.glide.ui.ng.cc</td>
<td>Catalog Designer Common.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.ng.dc, com.glideapp.workflow</td>
</tr>
</tbody>
</table>
| Certification Core            | com.snc.certification_core | core certification structures such as filters and templates. Certification Core cannot be activated by itself, but is activated automatically when any of these applications are activated:  
  • Desired State  
  • Architecture Compliance  
  • Data Certification  
  • IT Governance Risk and Compliance | inactive | false          | com.glide.list_v2, com.snc.version                                            |
<p>| Change Management - Collision Detector | com.snc.change.collision_detector | the ability to detect whether planned changes conflict with other changes, or have other scheduling issues. | active | true           | com.snc.maintenance_schedules                                                 |</p>
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Management - Core</td>
<td>com.snc.chan...</td>
<td>Core plugin with updates shared by Change Management State Model and Standard Change Catalog plugins.</td>
<td>active</td>
<td>true</td>
<td>com.snc.change_request</td>
</tr>
<tr>
<td>Change Management - Risk Assessment</td>
<td>com.snc.chan...</td>
<td>Risk assessment that is used to determine a risk based on data driven questions. This works in unison with with the condition-based change risk calculator.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.assessment, com.snc.bestpractice.task_survey, com.glideapp.survey, com.snc.bestpractice.change_risk_task_survey</td>
</tr>
<tr>
<td>Change Management - Standard Change Catalog</td>
<td>com.snc.chan...</td>
<td>Provides a way to propose, review, approve and create templates for Standard Changes.</td>
<td>true</td>
<td>true</td>
<td>com.snc.change_management, com.glideapp.servicecatalog</td>
</tr>
<tr>
<td>Change Management - State Model</td>
<td>com.snc.chan...</td>
<td>Provides a new state model for Normal, Emergency and Standard change requests. Activating will replace the current state model and workflows.</td>
<td>true</td>
<td></td>
<td>com.snc.change_management, com.snc.process_flow_formatter</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Change Management Workflows</td>
<td>com.glideapp.workflow_change_management</td>
<td>Adds three new template workflow versions which are designed for use with the change management process.</td>
<td>true</td>
<td>true</td>
<td>com.glideapp.workflow</td>
</tr>
<tr>
<td>Change Request</td>
<td>com.snc.change_request</td>
<td>Change Request.</td>
<td>active</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Chat</td>
<td>com.glideapp.live</td>
<td>Provides instant messaging, chat conversations, work queue, and group chat support.</td>
<td>inactive</td>
<td>true</td>
<td>com.glide.custom_web_service, com.glideapp.live_common, com.glide.db_audio</td>
</tr>
<tr>
<td>Chef Configuration Management</td>
<td>com.snc.chef</td>
<td>Provides support for the management of Chef environments for systems configuration management automation. Please consult with your ServiceNow representative.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.orchestration.activities.chef, Configuration Automation</td>
</tr>
<tr>
<td>Checklist</td>
<td>com.glide.ui.checklist</td>
<td>Provides a simple way to track the progress of tasks without creating additional records, using checklists that can be added to any form.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.ng, com.glide.rest.service</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Client Transaction Timings</td>
<td>com.glide.client</td>
<td>Provides support to track client rendering times at the server, lining up the values up with the server transaction times.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>CMS User Interface - Service Management Core</td>
<td>com.snc.service_management.core.cms</td>
<td>All CMS items (blocks, pages, and menus) used to reference our core IT self-service applications are packaged in this plugin. It is also the core foundation for all Service Management applications.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.cms, com.glide.cms.extensions, com.glide.db_images</td>
</tr>
<tr>
<td>Coaching Loops</td>
<td>com.snc.coaching_loops.application</td>
<td>Adds functionality that facilitates the coaching of employees on their work through the use of coaching opportunities that can be conditionally configured.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.planned_task, com.glideapp.custom_charts</td>
</tr>
<tr>
<td>Collaboration</td>
<td>com.glide.collaboration</td>
<td>Retired. Replaced by the Connect plugin.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.connect</td>
</tr>
<tr>
<td>Company extension</td>
<td>com.glide.company</td>
<td>Adds currency columns to the Company [core_company] table.</td>
<td>active</td>
<td>false</td>
<td>com.glide.currency</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Company Separation (Deprecated)</td>
<td>com.glide.separation</td>
<td>Creates a multi-tenant environment within the database.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Configuration Automation</td>
<td>com.snc.configuration_automation</td>
<td>Provides automated shared base views for integrations with configuration management products. Please consult with your ServiceNow representative.</td>
<td>inactive</td>
<td>false</td>
<td>Automation Applications Core, Core Automation</td>
</tr>
<tr>
<td>Configuration Management (CMDB Enterprise Edition)</td>
<td>com.snc.cmdb.enterprise</td>
<td>Provides core functionality for the configuration management database, including enterprise hardware and configuration item relationships.</td>
<td>active</td>
<td>false</td>
<td>com.snc.cmdb</td>
</tr>
<tr>
<td>Configuration Management (CMDB)</td>
<td>com.snc.cmdb</td>
<td>Provides core functionality for the configuration management database, including enterprise hardware and configuration item relationships.</td>
<td>active</td>
<td>true</td>
<td>com.glide.custom_web_service</td>
</tr>
<tr>
<td>Connect</td>
<td>com.glide.connector</td>
<td>Provides a real-time messaging platform that connects you to your coworkers, bypassing email and static documents.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.live_feed, com.glideapp.live_common, com.glide.ui.ng, com.glide.ui.angularui, com.glide.ui.ng.amb, com.glide.notification.push</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Connect Support New in Geneva</td>
<td>com.glide.connect.support</td>
<td>Support the Connect messaging platform and enables support agents to provide real-time assistance to end users, using queues.</td>
<td>inactive</td>
<td>true</td>
<td>com.glide.connect</td>
</tr>
<tr>
<td>Content Management</td>
<td>com.glide.cms.application</td>
<td>Allows administrators to create custom, branded, web front ends on top of an existing ServiceNow instance.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.home, com.glide.text_search, com.glide.db_images</td>
</tr>
<tr>
<td>Content Management Extended Types</td>
<td>com.glide.cms.types</td>
<td>An extension to Content Management that adds iFrames and Flash frames.</td>
<td>active</td>
<td>false</td>
<td>com.glide.cms.type.flash, com.glide.cms.type.iframe</td>
</tr>
<tr>
<td>Context Ranking</td>
<td>com.glide.sorting</td>
<td>Support for drag and drop lists and ranking dialog. Context ranking allows a user to sort a collection of records independently of the attributes of those records. Context Ranking is activated automatically with the SDLC - Scrum Process Pack plugin.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.list_v2</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Sensitive Help</td>
<td>com.glide.context_help</td>
<td>Provides a context-sensitive help system, providing links to specific help pages (either on the official ServiceNow product documentation site or any other website). These help pages can be linked to the list or form view of any table, or to the form view of a specific record in a table.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Contextual Search Service</td>
<td>com.snc.contextual_search</td>
<td>Provides contextual search services.</td>
<td>active</td>
<td>false</td>
<td>com.snc.application.json_service</td>
</tr>
<tr>
<td>Contextual Security</td>
<td>com.glide.role_management</td>
<td>Provides the flexibility and power to protect information by controlling read/write/create/delete authorization.</td>
<td>active</td>
<td>false</td>
<td>com.glide.db_view, com.glide.ui_page</td>
</tr>
<tr>
<td>Contextual Security Auditor</td>
<td>com.glide.acl.security Auditor</td>
<td>Available by request (Requires the Contextual Security plugin)</td>
<td>inactive</td>
<td>false</td>
<td>Contextual Security</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Contextual Security: Role Management Enhancements</td>
<td>com.glide.role_management.inh_count</td>
<td>Prevents duplicate entries in sys_user_has_role for inherited roles, based on the value of the inh_count column.</td>
<td>active</td>
<td>false</td>
<td>com.glide.role_management.inh_count</td>
</tr>
<tr>
<td>Contextual Security: Role Management Enhancements REST API</td>
<td>com.glide.role_management.inh_count_rest_api</td>
<td>Prevents duplicate entries in sys_user_has_role for inherited roles, based on the value of the inh_count column.</td>
<td>active</td>
<td>false</td>
<td>com.glide.role_management.inh_count_rest_api, com.glide.scripted_rest_services</td>
</tr>
<tr>
<td>Contract Management</td>
<td>com.snc.contract_management</td>
<td>Provides the ability to manage all types of contracts.</td>
<td>active</td>
<td>true</td>
<td>com.snc.asset_management</td>
</tr>
<tr>
<td>CORS support for REST API</td>
<td>com.glide.rest.cors</td>
<td>CORS support for REST API.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Cost Management New name in Geneva; formerly IT Cost Management</td>
<td>com.snc.cost_management.application</td>
<td>Fragment operating costs for configuration items and task-related activities, allocate the costs to business consumers, and compare actual allocations to planned budgets.</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Create knowledge from problem</td>
<td>com.snc.problem_kb.feature_set</td>
<td>Provides the ability to create knowledge articles from problems.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>CSS Theme support</td>
<td>com.glide.ui.themes.core</td>
<td>Provides support for CSS customizations to the user interface.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.ui.themes.core</td>
</tr>
<tr>
<td>CSS Theme support - UI 14</td>
<td>com.glide.ui.themes.core</td>
<td>Provides UI 14 themes.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.themes.core</td>
</tr>
<tr>
<td>CTI Softphone New in Geneva</td>
<td>com.snc.cti</td>
<td>Enables Twilio integration using Notify and OpenFrame to provide softphone functions and call center capabilities. These include make, receive phone calls, transfer, hold and mute. Applications like Customer Service and Incident Management provide demo workflows for CTI. Please re-activate respective applications where you require CTI demo workflows.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.matching_rule, com.sn_openframe, com.snc.notify, com.snc.notify.twilio</td>
</tr>
<tr>
<td>Currency support for the service catalog</td>
<td>com.glideapp.servicecatalog.currency</td>
<td>Enables the service catalog to support fully localized currencies for item prices and options.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.currency, com.glideapp.servicecatalog.servicecatalog.currency</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Custom Charts</td>
<td>com.glideapp.custom_charts</td>
<td>Allows you to create charts when you have a requirement that cannot be satisfied with the Reports application.</td>
<td>inactive</td>
<td>false</td>
<td>com.glideapp.report2</td>
</tr>
<tr>
<td>Customer Logon Plugin</td>
<td>com.snc.customer_logon</td>
<td><strong>Available by request</strong> Authorized Service-now employees to log on to customer's systems.</td>
<td>inactive</td>
<td>false</td>
<td>UI Pages (define HTML pages and their processing)</td>
</tr>
<tr>
<td>Customer Registration</td>
<td>com.snc.customer_registration</td>
<td><strong>Available by request</strong> ServiceNow.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Customer Service</td>
<td>com.sn_customerservice</td>
<td>New in Geneva Customer service enables you to provide service and support for your external customers using several communication channels, such as email, web, and telephone. A case is created to keep track of the issue reported or service requested, and assigned to groups or agents. Customer service agents in your organization work on the cases and resolve issues.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.asset_management, com.snc.state_flows, com.glide.connect.support, com.snc.cs_base, com.snc.skills_management, com.snc.state_flows, com.snc.assessment_core, com.snc.process_flow_formatter, com.snc.task_relations, com.snc.task_activity, com.snc.matching_rule, com.snc.resolutionshaper, com.snc.openframe</td>
</tr>
<tr>
<td>Customer Service CTI</td>
<td>com.snc.customer_service_cti</td>
<td><strong>Demo Data</strong> Service CTI demo data.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.customer_service, com.snc.cti</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Data Access Gateway</td>
<td>com.snc.da.gateway</td>
<td><strong>Data Access Gateways</strong></td>
<td>inactive</td>
<td>false</td>
<td>High Availability Cloning, Test status viewer</td>
</tr>
<tr>
<td>Data Archiving</td>
<td>com.glide.auxdb</td>
<td>Moves data that is no longer necessary for immediate day-to-day access from primary tables into a set of archive tables. Typical candidates include historical ITIL documents such as incidents which were closed last year; the functionality supports archiving of non-ITIL documents as well.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.db.replicate</td>
</tr>
<tr>
<td>Data Certification</td>
<td>com.snc.certification_v2</td>
<td>Enables field-level certification of data, either scheduled or on-demand.</td>
<td>inactive</td>
<td>true</td>
<td>com.glide.list_v2, com.snc.certification_core</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Data Lookup and Record Matching Support</td>
<td>com.glide.data</td>
<td>allows administrators to define rules that automatically set one or more field values when certain conditions are met. Note that this plugin completely replaces Priority Lookup. Any custom logic defined in the CalculatePriority business rule will need to be manually translated into the new priority data lookup definition.</td>
<td>active</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Data Lookup and Record Matching Support for Service Catalog</td>
<td>com.glide.data</td>
<td>allows administrators to perform data lookups for variables on service catalog item screens, on requested items, and on catalog tasks as a user fills out the values contained in variables.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.servicecatalog, com.glide.data_lookup</td>
</tr>
<tr>
<td>Data Policy 2</td>
<td>com.glide.data</td>
<td>defines mandatory or read-only requirements for table fields.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Data Structures</td>
<td>com.snc.datastructure</td>
<td>Provides element types: DataStructure and DataObject for flyweight data that can be stored internally as JSON and utilized via the DataStructure API.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Database Rotation</td>
<td>com.snc.db.rotation</td>
<td>Allows large tables to be broken up into smaller, more manageable tables.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Database Rotation with Default Tables</td>
<td>com.snc.db.rotation_default_tables</td>
<td>Adds database rotation functionality to default tables.</td>
<td>active</td>
<td>false</td>
<td>com.snc.db.rotation</td>
</tr>
<tr>
<td>Database Storage for Audio Files</td>
<td>com.glide.db_audio</td>
<td>Allows audio files to be uploaded and stored in the database, and referenced in HTML.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Database Storage for Images</td>
<td>com.glide.db_images</td>
<td>Allows images to be uploaded and stored in the database, and referenced in HTML.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Database Storage for Video Files</td>
<td>com.glide.db_video</td>
<td>Allows video files to be uploaded and stored in the database, and referenced in HTML.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database Views</td>
<td>com.glide.db_view</td>
<td>Database Views allows you to define table joins for reporting purposes.</td>
<td>active</td>
<td>false</td>
<td>com.glide.db_view, com.glide.metrics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Allows you to define table joins for reporting purposes.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataTables 1.1.0 Components</td>
<td>com.glide.ui.ng.datatables</td>
<td>DataTables 1.1.0 Components.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demand Management</td>
<td>com.snc.demand_management</td>
<td>Demand Management capturing the demand and provides tools to screen, assess, and prioritize it.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.process_flow_formatter, com.snc.assessment_core, com.snc.timeline_visualization, com.snc.bubblechart_workbench</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Depreciation</td>
<td>com.snc.depreciation</td>
<td>Core deprecation capabilities.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.tablet.theme, com.glide.ui.ng, com.glide.ui.themes.doctype</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designer Common</td>
<td>com.glide.ui.ng.dc</td>
<td>Provides common components required by designers such as the form designer and the quiz designer.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.tablet.theme, com.glide.ui.ng, com.glide.ui.themes.doctype</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desired State Certification</td>
<td>com.snc.certification</td>
<td>Evaluates records to see if they match a desired state, scheduled or on-demand.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.certification_core</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Diagnostic Monitoring</td>
<td>com.glide.monitor.diagnostics</td>
<td>Provides advanced diagnostic monitoring of each node in a ServiceNow instance.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Dictionary overrides</td>
<td>com.glide.dictionary.override</td>
<td>Allows specific Dictionary values to be overridden for extended table elements. For example, this plugin allows the default value for the Assigned To field in the Incident table to be different than the default value specified for the Assigned To field in the Task table.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Differ Test</td>
<td>com.glide.ui.differ_test</td>
<td>Comprehensive testing of the Differ classes</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.system_update_set_preview</td>
</tr>
<tr>
<td>Discovery</td>
<td>com.snc.discovery</td>
<td>Discovers the existence and configuration of web services, computers, printers, network gear and other devices on a customer’s networks</td>
<td>inactive</td>
<td>true</td>
<td>Discovery - IP Based, Discovery Components for Amazon Web Services</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Domain Support - Domain Extensions</td>
<td>com.glide.domain.msp_extensions.installer</td>
<td>Available by request</td>
<td>inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>E164 Compliant Phone Number</td>
<td>com.glide.phone_number</td>
<td><strong>Ensures</strong> that all necessary information for a phone number is included and properly formatted to successfully route an international call over a territory's public telephone network.</td>
<td>active</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>ECC Queue Retry Policy</td>
<td>com.glideapp.ecc_retry_policy</td>
<td>Define retry policies for messages queued in the ECC queue.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Email - OAUTH support for IMAP and SMTP</td>
<td>com.glide.email.oauth</td>
<td>Support for XOAUTH and XOAUTH2 email authentication.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.email_accounts, com.snc.platform.security.oauth</td>
</tr>
<tr>
<td>Email Accounts</td>
<td>com.glide.email_accounts</td>
<td>Enables you to define email accounts and settings in individual records.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Email Automatic User Creation</td>
<td>com.glide.email</td>
<td>Sets glide.email.create_userid_from_email=true so that when automatic user creation is enabled, the UserID of newly-created users matches the user’s email address. Also widens sys_user.user_name column to 100 bytes in order to accommodate longer UserIDs based on email addresses.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Email Filters</td>
<td>com.glide.email</td>
<td>Filters emails into different mailboxes or junk, depending on headers and subject. Ignores any email that contains a VCAL invitation.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Email Notification Preview Plugin</td>
<td>com.glide.email</td>
<td>New in Geneva. Allows you to easily preview a ServiceNow email notification without sending the notification. This allows you to preview notifications at design time.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email/SMS Separation</td>
<td>com.glide.email_sms_separation</td>
<td>Separation of the sending of SMS and email by adding a column to the email table. Prevents SMS messages from slowing down email message sending and vice versa. Activation of this plugin will cause email to stop sending during activation. On systems with large email tables this can take hours, and is not recommended.</td>
<td>active</td>
<td>false</td>
<td>com.glide.notification</td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encryption Support</td>
<td>com.glide.encryption</td>
<td>Allows text fields and attached files to be encrypted.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Encryption Support - Single Context Task Encryption Demo</td>
<td>com.snc.task_encryptiondemo</td>
<td>Cryptos showing how encrypted form fields can be assigned to one encryption context. Adds two encrypted fields to the Task table (Social Security Numbers and Credit Cards), a read-only encryption context field, an <strong>Add Secure Info</strong> button that unhides these fields, and a <strong>Change Encryption</strong> button to change the encryption context. After activating, configure the Incident form to add the three fields mentioned above.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.encryption</td>
</tr>
<tr>
<td>Encryption Wrapper - Database</td>
<td>com.glide.encryption_wrapper</td>
<td>Allows encryption key to be placed in a DataSecure network attached encryption device.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.encryption_wrapper</td>
</tr>
<tr>
<td>Encryption Wrapper - Network Attached Encryption</td>
<td>com.glide.encryption_wrapper.nae</td>
<td>Allows encryption key to be placed in a DataSecure network attached encryption device.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.encryption_wrapper</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption Wrapper Common Items</td>
<td>com.glide.encryption_wrapper.common</td>
<td>Items common to encryption wrappers.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Engine Based Notifications</td>
<td>com.glide.email_engine_notifs</td>
<td>Container for engine-based email notifications. Contains a set of default email notifications. Installed only on new z-boots.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Enhanced Web Service Provider - Common</td>
<td>com.glide.web_service_provider_v2</td>
<td>ESS portal content management application. Demo data includes the actual ESS portal.</td>
<td>active</td>
<td>true</td>
<td>com.glide.cms</td>
</tr>
<tr>
<td>ESS Portal (implemented within Content Management)</td>
<td>com.glide.cms.extensions</td>
<td>ESS portal content management application. Demo data includes the actual ESS portal.</td>
<td>active</td>
<td>true</td>
<td>com.glide.cms</td>
</tr>
<tr>
<td>Event Management Available by request</td>
<td>com.glideapp.itom.snac.event_management</td>
<td>Service oriented availability management and event console.</td>
<td>inactive</td>
<td>true</td>
<td>Event Management and Service Mapping Core, Timeline Visualization</td>
</tr>
<tr>
<td>Event Management and Service Mapping Core New in Geneva</td>
<td>com.snc.service_watch</td>
<td>Common components of Event Management and Service Mapping.</td>
<td>inactive</td>
<td></td>
<td>Discovery - IP Based, Service Modeling, REST API Provider</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Plan support for the service catalog</td>
<td>com.glideapp.servicecatalog.execution_plan</td>
<td>New in Geneva Supports execution plans of the Service Catalog.</td>
<td>active</td>
<td>true</td>
<td>com.glide.execution_plan</td>
</tr>
<tr>
<td>Expense Line</td>
<td>com.snc.expense_line</td>
<td>Core expense line table that enables cost tracking. Integrated with asset management, CMDB, cost management, and contract management.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Explicit Roles</td>
<td>com.glide.explicit_roles</td>
<td>Do explicit role checks. Users and ACLs are required to have roles (Requires Contextual Security)</td>
<td>inactive</td>
<td></td>
<td>Contextual Security Rules</td>
</tr>
<tr>
<td>Extended CMDB</td>
<td>com.snc.extended_cmdb</td>
<td>Provides specialized configuration items, such as radio hardware, test equipment, and voice system hardware.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.cmdb</td>
</tr>
<tr>
<td>External Credential Storage</td>
<td>com.snc.discovery.external_credentials</td>
<td>Available by request Provides the credentials used by Discovery and Orchestration to be stored on the MID Server rather than the instance.</td>
<td>inactive</td>
<td></td>
<td>MID Server</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Facilities Service Management</td>
<td>com.snc.facilities_service_automation</td>
<td>Manages facilities requests and enables users to report and track requests by their location on a floor plan. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_management</td>
</tr>
<tr>
<td>Facilities Service Management CMS Portal</td>
<td>com.snc.facilities_service_automation.cms</td>
<td>Manages facilities requests and enables users to report and track requests by their location on a floor plan. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.facilities_service_automation, com.snc.service_management.cms</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Facilities Service Management Floor Plan Viewer</td>
<td>com.snc.facilities_service_automation.fpv</td>
<td>The facilities legacy floor plan viewer is image-based. This feature has been replaced by the interactive facility maps application.</td>
<td>active</td>
<td>true</td>
<td>com.snc.facilities_service_automation, com.glide.ui.angular, com.glide.ui.font_icons</td>
</tr>
<tr>
<td>Facilities Visualization Workbench</td>
<td>com.snc.facilities_service_automation.fvw</td>
<td>The Facilities Visualization Workbench provides map interaction built upon geoJSON file map sets.</td>
<td>active</td>
<td>true</td>
<td>com.snc.facilities_service_automation, com.snc.facilities_service_automation.move</td>
</tr>
<tr>
<td>Field Normalization</td>
<td>com.snc.field_normalization</td>
<td>Provides support for cleaning up messy data through normalization and transformation.</td>
<td>inactive</td>
<td>true</td>
<td>com.glide.vars, com.glide.system_update_set</td>
</tr>
<tr>
<td>Field Service Management</td>
<td>com.snc.work_management</td>
<td>Lets you manage on-location work tasks of any kind.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.service_management</td>
</tr>
<tr>
<td>Field Service Management CMS Portal</td>
<td>com.snc.work_management.cms</td>
<td>Lets you launch Field Service Automation and other service management applications from a single CMS page.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.work_management, com.snc.service_management</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Field Service Management Demo Data</td>
<td>com.snc.work_management.demo</td>
<td>Demonstration Data for Field Service Management covering the medical and telecommunication domains. NOTE: Installing this plugin will add new Configuration Item tables and relationships to the database.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.work_management.demo</td>
</tr>
<tr>
<td>Field Service Management Geolocation Demo Data</td>
<td>com.snc.work_management_geolocation.demo</td>
<td>Field Service Geolocation capabilities.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.work_management_geolocation.demo</td>
</tr>
<tr>
<td>Field Service Management Mobile</td>
<td>com.snc.work_management.m</td>
<td>Manages new mobile UI for Field Service. This plugin will automatically install or upgrade the Field Service and Mobile UI plugins.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.service_management.m, com.snc.work_management.m</td>
</tr>
<tr>
<td>Finance Service Management</td>
<td>com.snc.finance_service_automation</td>
<td>Manages finance requests and enables users to report and track those requests. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_management.m, com.snc.work_management.m</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance Service Management CMS Portal</td>
<td>com.snc.finance_service_automation.cms</td>
<td>Lets you launch Finance Service Automation and other service management applications from a single CMS page. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.finance_service_automation, com.snc.service_management.cms</td>
</tr>
<tr>
<td>Financial Management New name in Geneva; formerly IT Finance Finance</td>
<td>com.snc.financial_management.application</td>
<td>Enables financial analysts to assemble spending data, build cost models, and generate reports to show how funds are being used. Activation of this plugin on production instances may have licensing implications. Contact your ServiceNow account team for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.it_data_mart, com.snc.fiscal_calendar, com.snc.project_portfolio_suite, com.snc.common_workbench, com.glide.ui.ng, com.glide.ui.expanding_iframe</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Financial Management with Performance Analytics</td>
<td>com.snc.financial_management_with_performance_analytics</td>
<td>Use of Performance Analytics to build unit cost reporting with data from Financial Management. Activation of either Performance Analytics or Financial Management in production instances may have licensing implications. Contact your ServiceNow account team for details.</td>
<td>inactive</td>
<td>false</td>
<td>Financial Management, Performance Analytics</td>
</tr>
<tr>
<td>Fiscal Calendar</td>
<td>com.snc.fiscal_calendar</td>
<td>Provides support for creating and managing fiscal calendars.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.dynamic_operands.datetime</td>
</tr>
<tr>
<td>Fixed Asset</td>
<td>com.snc.fixed_asset</td>
<td>Sets fixed asset tracking.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.ng.dc, com.glide.ui.tablet.theme, com.glide.ui.ng, com.glide.ui.themes.doctype</td>
</tr>
<tr>
<td>Form Designer</td>
<td>com.glide.ui.ng.fd</td>
<td>Form Designer.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.doctype</td>
</tr>
<tr>
<td>Form Personalization</td>
<td>com.glide.ui.personalize_form</td>
<td>Enables users to personalize the layout for any form view.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.doctype</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Geolocation</td>
<td>com.snc.geolocation.application</td>
<td>Uses Google Maps to track users, plan efficient routes between locations, and assist in finding accurate travel times. The system locates users from latitude and longitude information provided by their mobile devices.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
</table>
| Glide Core                    | com.glide.core     | Core plugin for the glide platform.                                         | inactive| false          | Glide Core, Script Whitelist Manager - Prevent Collection Mode, Schema for contextual security, Schema for ui features, Form Annotations, UI Pages (define HTML pages and their processing), UI Policy Rules, Reporting, Table Cleaner Upgrade Plugin, Dictionary overrides, View Management, Cluster Plugin, Automated Action Interface, System Mailboxes, Plugin Management, Basic Import/Export Functionality, Dashboards, com.glide.instance_registration_client, System Update Sets (viewer), Database Storage for Images, Request Manager (transaction cancel), Semaphore Management, com.glide.system_property_categories, History Sets, Lists v2, com.glide.security.strict_read_roles, Version Support, ...

© 2017 ServiceNow. All rights reserved. 1268
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glide Metadata</td>
<td>com.glide.metadata</td>
<td>Core metadata support.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Glide Metadata Delete</td>
<td>com.glide.metadata</td>
<td>Core metadata delete support.</td>
<td>active</td>
<td>false</td>
<td>com.snc.metadata, com.snc.metadata_tree</td>
</tr>
<tr>
<td>Global Trends for Text Search</td>
<td>com.glide.ts_global</td>
<td>Global Trends for Text Search.</td>
<td>inactive</td>
<td>false</td>
<td>Zing Text Search</td>
</tr>
<tr>
<td></td>
<td>ts_global_trends</td>
<td>Available by request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Goal And Objective Management</td>
<td>com.snc.goals_and</td>
<td>application provides functionality for managing goals and objectives</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td></td>
<td>goals_and_objectives</td>
<td>review derived from records and metrics within a Service-Now instance.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>application</td>
<td>Available by request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Custom Search Integration</td>
<td>com.snc.integration</td>
<td>integration with the Integration team prior to activating this plugin.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td></td>
<td>google_custom_search</td>
<td>Available by request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Maps Plugin</td>
<td>com.glideapp.google</td>
<td>display of Google maps within the product as map pages.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.ng, com.glide.rest.service</td>
</tr>
<tr>
<td></td>
<td>google_maps</td>
<td>Available by request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Governance, Risk, and Compliance (GRC)</td>
<td>com.snc.governance</td>
<td>Governance, Risk, and Compliance application.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.governance_core, com.snc.certification_core, com.snc.assessment_core</td>
</tr>
<tr>
<td></td>
<td>application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GRC: Risk New in Geneva</td>
<td>com.sn_risk</td>
<td>Allows organizations to identify, assess, and respond to risk throughout the enterprise. Activation of this plugin on production instances may have licensing implications. Contact your ServiceNow account team for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.sn_grc, com.snc.governance_core, com.snc.certification_core, com.snc.assessment_core</td>
</tr>
<tr>
<td>High Security Settings</td>
<td>com.glide.high_security</td>
<td>Enables a high-security environment.</td>
<td>active</td>
<td>false</td>
<td>com.glide.role_management</td>
</tr>
<tr>
<td>History Sets</td>
<td>com.glide.history</td>
<td>Maintains sys_history_set and sys_history_line tables to view a record's audit, email, and relationship data in a table format.</td>
<td>active</td>
<td>false</td>
<td>com.glide.history</td>
</tr>
<tr>
<td>Homepage Splash Page</td>
<td>com.glideapp.home_splash_page</td>
<td>Provides a splash page for homepages. Upon logging in, instead of going to home.do, users will go to a splash page that gives them the ability to cancel the home page transaction. This is useful when homepages take a long time to load.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.home</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTML Sanitizer</td>
<td>com.glide.htmlsanitizer</td>
<td>Automatically cleans up HTML markup in all HTML fields to remove unwanted code in order to protect against security concerns such as cross-site scripting attacks.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Human Resources Application:</td>
<td>com.snc.hr.core</td>
<td>Lets you automate standardized human resources (HR) processes within your organization, including handling changes to employee status. Requires a subscription. In Geneva, one plugin includes Core and Employee Change.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_management, com.glide.data_lookup, com.glide.country, com.glide.cms, com.glide.cms.extensions, com.glide.db_images, com.snc.pdf_generator</td>
</tr>
<tr>
<td>Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New in Geneva</td>
<td>com.snc.hr.hr_connect</td>
<td>Installs the Chat with HR menu item and HR chat queue supporting files.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.hr.core, com.glide.connect, com.glide.connect.support</td>
</tr>
<tr>
<td>Human Resources Application:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HR Connect</td>
<td>com.snc.hr.pa</td>
<td>Provides Performance Analytics content pack for HR. Requires a subscription to Performance Analytics.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.hr.core, com.snc.pa.premium, com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Resources Application: Workday Integration</td>
<td>com.snc.hr.workday.application</td>
<td>Provides automatic updates of HR application tables from a linked Workday instance. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.hr.core</td>
</tr>
<tr>
<td>I18N: Brazilian Portuguese Translations</td>
<td>com.snc.i18n.brazilian_portuguese.feature_set</td>
<td>Internationalization plugin for language internationalization. Provides Brazilian Portuguese.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Czech Translations</td>
<td>com.snc.i18n.czech.feature_set</td>
<td>Internationalization plugin for language internationalization. Provides Czech.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Dutch Translations</td>
<td>com.snc.i18n.dutch.feature_set</td>
<td>Internationalization plugin for language internationalization. Provides Dutch.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Finnish Translations</td>
<td>com.snc.i18n.finnish.feature_set</td>
<td>Internationalization plugin for language internationalization. Provides Finnish.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>118N: French - Canada Translations</td>
<td>com.snc.i18n.french-canada</td>
<td>Internationalization plugin for language internationalization. Provides French - Canadian.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>118N: French Translations</td>
<td>com.snc.i18n.french</td>
<td>Internationalization plugin for language internationalization. Provides French.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>118N: German Translations</td>
<td>com.snc.i18n.german</td>
<td>Internationalization plugin for language internationalization. Provides German.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>118N: Hebrew Translations</td>
<td>com.snc.i18n.hebrew</td>
<td>Internationalization plugin for language internationalization. Provides Hebrew.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>118N: Hungarian Translations</td>
<td>com.snc.i18n.hungarian</td>
<td>Internationalization plugin for language internationalization. Provides Hungarian.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>118N: Internationalization</td>
<td>com.glide.i18n</td>
<td>An Internationalization plugin for language internationalization. Provides the elements necessary for translating an instance without any translation preloaded.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.system_import_set, com.glideapp.knowledge.i18n</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>I18N: International Translation helper</td>
<td>com.glide.i18n.translation_helper</td>
<td>Mobile and import set maps to merge all languages into one table for translation maintenance.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: International Translation Verification</td>
<td>com.glide.i18n.translation_verification</td>
<td>Development Modules, UI Pages, Scripts to load translation files for verification</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n.translation_helper</td>
</tr>
<tr>
<td>I18N: Italian Translations</td>
<td>com.snc.i18n.italian</td>
<td>An Internationalization plugin for language internationalization. Provides Italian.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Japanese Translations</td>
<td>com.snc.i18n.japanese</td>
<td>An Internationalization plugin for language internationalization. Provides Japanese.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Knowledge Management Internationalization Plugin v2</td>
<td>com.glideapp.knowledge.i18n2</td>
<td>Translation of knowledge articles, and some UI enhancements to the knowledge base.</td>
<td>inactive</td>
<td>false</td>
<td>com.glideapp.knowledge3, com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Korean Translations</td>
<td>com.snc.i18n.korean</td>
<td>An Internationalization plugin for language internationalization. Provides Korean.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>I18N: Polish Translations</td>
<td>com.snc.i18n.polish</td>
<td>Internationalization plugin for language internationalization. Provides Polish.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Portuguese Translations</td>
<td>com.snc.i18n.portuguese</td>
<td>Internationalization plugin for language internationalization. Provides Portuguese.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Russian Translations</td>
<td>com.snc.i18n.russian</td>
<td>Internationalization plugin for language internationalization. Provides Russian.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Simplified Chinese Translations</td>
<td>com.snc.i18n.chinese</td>
<td>Internationalization plugin for language internationalization. Provides Simplified Chinese.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Spanish Translations</td>
<td>com.snc.i18n.spanish</td>
<td>Internationalization plugin for language internationalization. Provides Spanish.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Thai Translations</td>
<td>com.snc.i18n.thai</td>
<td>Internationalization plugin for language internationalization. Provides Thai.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>I18N: Traditional Chinese Translations</td>
<td>com.snc.i18n.traditional_chinese</td>
<td>Internationalization plugin for language internationalization. Provides Traditional Chinese.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>I18N: Turkish Translations</td>
<td>com.snc.i18n.turk</td>
<td>Internationalization plugin for language internationalization. Provides Turkish.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.i18n</td>
</tr>
<tr>
<td>Incident Alert Management</td>
<td>com.snc.iam</td>
<td>Allows crisis managers to manage communications for major issues, to bring together all involved users and help resolve these issues quickly.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.contact_management, com.glideapp.live_feed_document, com.glide.phone_number</td>
</tr>
<tr>
<td>Incident Management</td>
<td>com.snc.incident</td>
<td>Restores normal service operation as quickly as possible following an incident, while minimizing impact to business operations and ensuring quality is maintained.</td>
<td>active</td>
<td>true</td>
<td>com.snc.service</td>
</tr>
<tr>
<td>Incident Resolution Fields</td>
<td>com.snc.incident_resolution_fields</td>
<td>Adds Resolved and Resolved by fields to the Incident table, similar to Closed and Closed by, populated with a business rule when an incident is resolved or closed.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Insert Multiple Web Service</td>
<td>com.glide.web_service_insert_multiple</td>
<td>Enables multiple inserts for the Direct SOAP API.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Integration - Altiris 2.0</td>
<td>com.snc.integration.altiris2</td>
<td>integration with Altiris.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.integration.jdbc</td>
</tr>
<tr>
<td>Integration - Common Components</td>
<td>com.snc.integration.common</td>
<td>common scripts for integrations.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.system_import_set, com.glide.web_service_provider, com.glide.system_property_categories</td>
</tr>
<tr>
<td>Integration - LANDesk (Deprecated)</td>
<td>com.snc.integration.landesk</td>
<td>integration with LANDesk.</td>
<td>inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration - Microsoft SCCM (Deprecated)</td>
<td>com.snc.integration.sccm</td>
<td>version 3.0 integration with Microsoft SCCM. Replaced by Integration - Microsoft SCCM 2007.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.integration.jdbc</td>
</tr>
<tr>
<td>Integration - Microsoft SMS Available by request</td>
<td>com.snc.integration.sms</td>
<td>integration with Microsoft SMS</td>
<td>inactive</td>
<td>false</td>
<td>Microsoft SMS Import Maps, Monitor Plugin, Integration - Common Components</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Integration - Microsoft SMS / SCCM 2.0 (Deprecated)</td>
<td>com.snc.integration.sms2</td>
<td>2.0 integration with Microsoft SMS/SCCM. Replaced by Integration - Microsoft SCCM 2007.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.integration.jdbc</td>
</tr>
<tr>
<td>Integration - Microsoft System Center Operations Manager (SCOM)</td>
<td>com.snc.integration.scom</td>
<td>Integration with Microsoft System Center Operations Manager</td>
<td>inactive</td>
<td>false</td>
<td>Integration - Common Components, com.glide.web_service_application, MID Server</td>
</tr>
<tr>
<td>Integration - Multifactor Authentication</td>
<td>com.snc.integration.multifactor.authentication</td>
<td>Activate this plugin to set up Multifactor authentication on the instance.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Integration - Multiple Provider Single Sign-On Installer</td>
<td>com.snc.integration.sso.multi.installer</td>
<td>Use this plugin instead of the Integration - Multiple Provider Single Sign-On plugin to activate the Multiple Provider Single Sign-On feature.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Integration - OpenID SSO</td>
<td>com.snc.integration.openid</td>
<td>Enables single sign-on by exchanging URL parameters with an external OpenID Provider (OP).</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.integration.common</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IP Range Based Authentication</td>
<td>com.snc.ipauthenticator</td>
<td>Controls access to an instance based on IP address.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>IT Data Mart</td>
<td>com.snc.it_data_application</td>
<td>Stores the information that the IT Finance application uses to allocate expenses to specific accounts and segments in the general ledger.</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Item Designer support for the service catalog</td>
<td>com.glideapp.servicecatalog.item_design</td>
<td>Service Catalog item designer.</td>
<td>true</td>
<td></td>
<td>com.snc.incident, com.snc.problem, com.snc.change_request, com.snc.sla, com.snc.cmdb, com.snc.service_portfolio, com.snc.service_portfolio.sla</td>
</tr>
<tr>
<td>ITSM and PA Demo Data</td>
<td>com.snc.itsm_pa_demo</td>
<td>Demo data for Incident, Problem, Change, Task SLAs, Business Services, Service Offering, Service Commitments and Performance Analytics.</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>JavaScript Editor</td>
<td>com.glide.javascript_editor</td>
<td>Enables syntax highlighting and script macros for JavaScript fields.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>JSON Service request/response model</td>
<td>com.snc.application.json_service</td>
<td>Does JSON components and helpers for a JSON request/response model. Includes JSON and XML transports for NG and GlideAjax support. An extension of the processor framework.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Kepner Tregoe - Structured Problem Analysis</td>
<td>com.snc.kt_pa_application</td>
<td>The KT Structured Problem Analysis application uses the &quot;Kepner-Tregoe&quot; Problem Analysis methodology, to help investigate the root cause of a problem.</td>
<td>inactive</td>
<td>false</td>
<td>Coaching Loops</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Keylines Business Service Maps</td>
<td>com.snc.keylines.application</td>
<td>Provides an interactive and graphical interface to visualize configuration items (CIs) and their relationships. Provides filtering capabilities to manage data being displayed, allowing for configuration by the user to view in context to their role. Additional capabilities provide for the displaying of related tasks such as incidents, problems, changes, and certification tasks.</td>
<td>active</td>
<td>false</td>
<td>com.glide.diagrammer, com.snc.keylines, com.glideapp.bsm_map2</td>
</tr>
<tr>
<td>Keylines Core</td>
<td>com.snc.keylines.application</td>
<td>Contains the Keylines Javascript library for drawing diagrams.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Knowledge Document</td>
<td>com.snc.knowledge_document feature_set</td>
<td>Adds knowledge based functionalities to the Managed Documents plugin. You can create a knowledge article from a document, or update a knowledge document to a newer revision.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.document_management</td>
</tr>
<tr>
<td>Knowledge Management for Express</td>
<td>com.glide.express.knowledge</td>
<td>Knowledge plugin for Express.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.knowledge3</td>
</tr>
<tr>
<td>Knowledge Management V3</td>
<td>com.snc.knowledge3</td>
<td></td>
<td>active</td>
<td>true</td>
<td>com.glide.ui.angular, com.glide.ui.ng, com.glideapp.knowledge2, com.glideapp.user_criteria, com.snc.contextual_search, com.snc.db.rotation, com.glideapp.live_feed_v2</td>
</tr>
<tr>
<td>Knowledge Management Wiki Support</td>
<td>com.glideapp.knowledge2.wiki</td>
<td></td>
<td>active</td>
<td>false</td>
<td>com.glide.wiki, com.glideapp.knowledge2</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------</td>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Legacy Notify</td>
<td>com.snc.notifynow</td>
<td>Notify enables bi-directional notifications over SMS, Voice and Conference Bridges. Tightly integrated with the workflow engine and business rules, it delivers a highly configurable and trusted way to deliver messages across a range of channels. Requires a separate contract with Twilio for SMS and Voice capabilities.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.notification, com.glide.phone_number</td>
</tr>
<tr>
<td>Legal Service Management</td>
<td>com.snc.legal_service_management</td>
<td>Manages legal matters and enables users to report and track matters. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_management</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Legal Service Management CMS Portal</td>
<td>com.snc.legal_service_automation.cms</td>
<td>Lets you launch Legal Service Automation and other service management applications from a single CMS page. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>active</td>
<td>false</td>
<td>com.snc.legal_service_automation, com.snc.service_management.core.cms</td>
</tr>
<tr>
<td></td>
<td>com.snc.legal_service_automation.m</td>
<td>Legal Service Management mobile components.</td>
<td>active</td>
<td>false</td>
<td>com.snc.legal_service_automation, com.snc.service_management.core.m</td>
</tr>
<tr>
<td></td>
<td>com.glide.list_v2</td>
<td>Updates to the display of lists that include a cleanup UI, hierarchical lists and related lists embedded in forms.</td>
<td>active</td>
<td>false</td>
<td>com.glide.tiny_url, com.glide.ui_list_edit_with_form, com.glide.db_context_menu</td>
</tr>
<tr>
<td></td>
<td>com.glideapp.live_feed</td>
<td>Provides a place to post and share content in a ServiceNow instance.</td>
<td>active</td>
<td>true</td>
<td>com.glide.custom_web_service, com.glideapp.live_common, com.glide.notification</td>
</tr>
<tr>
<td></td>
<td>com.glideapp.live_feed_v2</td>
<td>Provides an updated application to post and share content in a ServiceNow instance.</td>
<td>active</td>
<td>true</td>
<td>com.glideapp.live_feed, com.glideapp.ui_components</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Live Feed Document -</td>
<td>com.glideapp.live_feed_document</td>
<td>Enables you to manage your task conversations and comments from My Feed or a document group.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.live_feed</td>
</tr>
<tr>
<td>follow tasks (Incident,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change, etc)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magellan Navigator</td>
<td>com.glide.ui.magellan_navigator</td>
<td>Provides a redesigned application navigator for UI16. Combines standard navigation capabilities, customizable favorites, and recently accessed items in a single responsive control.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.magellan_navigator_api, com.glide.ui.doctype, com.glide.ui.ng</td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance Schedules</td>
<td>com.snc.maintenance_schedules</td>
<td>Links configuration items to maintenance schedules. The maintenance schedules are checked against the planned dates for changes, and those that appear outside the maintenance schedule are so marked.</td>
<td>active</td>
<td>false</td>
<td>com.glide.schedules, com.snc.cmdb</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Managed documents</td>
<td>com.snc.document_management</td>
<td>Lightweight, ITIL-based solution for managing electronic documents within your ServiceNow instance. To enable the ability to publish to the knowledge base, activate the Knowledge Document plugin.</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Marketing Service Management</td>
<td>com.snc.marketing_service_automation</td>
<td>Manages marketing requests and enables users to report and track those requests. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_management</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Marketing Service Management CMS Portal</td>
<td>com.snc.marketing_service_automation.cms</td>
<td>lets you launch Marketing Service Automation and other service management applications from a single CMS page. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>active</td>
<td>true</td>
<td>com.snc.marketing_service_automation, com.snc.service_management.cms</td>
</tr>
<tr>
<td>Marketing Service Management Mobile</td>
<td>com.snc.marketing_service_automation_mobile</td>
<td>Manages Marketing Service Management mobile components.</td>
<td>active</td>
<td>false</td>
<td>com.snc.marketing_service_automation, com.snc.service_management.core.cms</td>
</tr>
<tr>
<td>Merge Tool</td>
<td>com.glide.ui.merge Tool</td>
<td>User interface for performing merges between two payloads.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.angular, com.glide.snc_code_editor</td>
</tr>
<tr>
<td>Metadata Tree</td>
<td>com.snc.metadata_tree</td>
<td>Hierarchical representation of metadata.</td>
<td>active</td>
<td>false</td>
<td>com.snc.apps_file</td>
</tr>
<tr>
<td>Metric Definition</td>
<td>com.glide.metrics</td>
<td>Provides an easy, declarative way to define metrics and allows the definitions to be tracked and stored by the system.</td>
<td>active</td>
<td>true</td>
<td>com.glide.schedules</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Microsoft Azure</td>
<td>com.sn_azure application</td>
<td>Plugin for the integration with Microsoft Azure.</td>
<td>inactive</td>
<td></td>
<td>com.snc.discovery.web_service.core, com.snc.runbook_automation.virtualization</td>
</tr>
<tr>
<td>MID Server</td>
<td>com.glideapp.agent</td>
<td>Management, Instrumentation, and Discovery (MID) Server is a Java application that runs as a Windows service or UNIX daemon. The MID Server facilitates communication and movement of data between the ServiceNow platform and external applications, data sources, and services.</td>
<td>active</td>
<td>false</td>
<td>com.glide.custom_web_service</td>
</tr>
<tr>
<td>Mobile Device ITIL and Service Management</td>
<td>com.snc.itil_mobile_app</td>
<td>Applications and modules for ITIL and Service Management.</td>
<td>active</td>
<td>false</td>
<td>com.glide.labels, com.glide.mobile</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Model Management</td>
<td>com.snc.model</td>
<td>Enables you to manage and maintain model categories, models, suites and bundled models.</td>
<td>active</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>My Assets</td>
<td>com.snc.asset_myassets</td>
<td>Provides users with self-service access to their own assets, contracts, and requests.</td>
<td>active</td>
<td>false</td>
<td>com.snc.asset_management</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Next-Gen BSM</td>
<td>com.snc.ng_bsm</td>
<td>Next Generation BSM (NG-BSM) built on D3 and Angular. Provides an enhanced, modern interactive graphical interface to visualize Configuration Items (CIs) and their relationships. Provides filtering capabilities to manage data being displayed and displays related information for CIs such as Events, Incidents, Problems and Changes.</td>
<td>active</td>
<td>false</td>
<td>com.snc.keylines_bsm_map, com.glide.ui.heisenberg</td>
</tr>
<tr>
<td>NG shared components</td>
<td>com.glide.ui.ng</td>
<td>Provides libraries and services common to plugins using Angular.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.angular, com.glide.ui.angularui, com.glide.ui.ng.amb</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Notify</td>
<td>com.snc.notify</td>
<td>Notify provides powerful platform features for workflow-driven voice calls,</td>
<td>inactive</td>
<td>true</td>
<td>com.glide.phone_number,</td>
</tr>
<tr>
<td></td>
<td>feature_set</td>
<td>conference calls, and SMS messages making it possible to create flexible</td>
<td></td>
<td></td>
<td>com.snc.notify.twilio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interactive Voice Response (IVR) systems to do virtually anything. Requires</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the Twilio Driver and a separate contract with Twilio for SMS and Voice</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>capabilities.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notify - Twilio Driver</td>
<td>com.snc.notify.twilio</td>
<td>Notify support for Twilio. Requires a separate contract with Twilio for SMS</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.phone_number</td>
</tr>
<tr>
<td></td>
<td>feature_set</td>
<td>and Voice capabilities.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OAuth 2.0</td>
<td>com.snc.platform.security.oauth</td>
<td>The implementation of OAuth 2.0 to support token granting and authentication.</td>
<td>active</td>
<td>false</td>
<td>com.glide.certificates,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>com.snc.platform.security.oauth.legacy</td>
</tr>
<tr>
<td>OAuth 2.0 legacy</td>
<td>com.snc.platform.security.oauth</td>
<td>Legacy implementation of OAuth 2.0. Please instead install com.snc.platform.security.oauth.</td>
<td>legacy</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>On-Call Scheduling</td>
<td>com.snc.on_call</td>
<td>Provides the ability to create on-call schedules and escalation trees. When an incident is created, dynamically route the escalation to an on-call resource. On Call allows you to configure and build different on-call schedules per process and assignment group. When utilizing the Notify plugin, resources can use SMS and Voice escalations to interact with the the escalation to acknowledge incidents, etc.</td>
<td>inactive</td>
<td>true</td>
<td>com.glide.schedules, com.glide.notification</td>
</tr>
<tr>
<td>Openframe</td>
<td>com.sn_openframe</td>
<td>An interface to integrate external communication systems with ServiceNow. This plugin brings a UI frame that is accessible and available anywhere within ServiceNow screens.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.ui.ng</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orchestration</td>
<td>com.snc.runbook_automation</td>
<td><strong>Available as a separate subscription</strong> Automation of oft-repeated manual operations such as deployments, configurations, information gathering, etc. Uses workflow and MID server technologies to allow activities to execute commands on equipment inside the customer’s enterprise network.</td>
<td>inactive</td>
<td></td>
<td>Workflow Runtime Engine, Core Automation, Orchestration - ROI</td>
</tr>
<tr>
<td>Orchestration Virtualization</td>
<td>com.snc.runbook_automation.virtualization</td>
<td><strong>Available by request</strong> Orchestration Virtualization features for various Cloud/ Virtualization hypervisors.</td>
<td>inactive</td>
<td></td>
<td>Orchestration, Automation Applications Core, Orchestration Virtualization Core Components</td>
</tr>
<tr>
<td>Orchestration - Active Directory</td>
<td>com.snc.orchestration.ad</td>
<td><strong>Available by request</strong> Orchestration activity pack for Active Directory</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration - Amazon EC2</td>
<td>com.snc.ec2_v2</td>
<td><strong>Available by request</strong> Support for provisioning Amazon EC2 instances from images (AMIs). Includes a service catalog entry for ordering an instance running Windows or Linux.</td>
<td>inactive</td>
<td></td>
<td>com.snc.orchestration.activities.ec2, com.snc.runbook_automation.virtualization, Amazon Web Services Common, Discovery Components for Amazon Web Services</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Orchestration - Amazon EC2 Integration for Training Partners</td>
<td>com.snc.ec2_training_partner</td>
<td>An application that ServiceNow's partners that will provide training can create lab instances using their own Amazon EC2 account, using images shared by ServiceNow's training department.</td>
<td>inactive</td>
<td></td>
<td>Orchestration - Amazon EC2</td>
</tr>
<tr>
<td>Available by request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orchestration - Examples</td>
<td>com.snc.runbook_automation.examples</td>
<td>Various examples of how Orchestration can be used.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Available by request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orchestration - Exchange</td>
<td>com.snc.orchestration.exchange</td>
<td>The Exchange activity pack.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration - F5 network management</td>
<td>com.snc.orchestration.f5</td>
<td>The activity pack for F5.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration - Infoblox DDI Activity Pack</td>
<td>com.snc.orchestration.infoblox</td>
<td>The Infoblox activity pack.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration - PowerShell</td>
<td>com.snc.orchestration.powershell</td>
<td>The activity pack for PowerShell.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration - ROI</td>
<td>com.snc.runbook_automation.roi</td>
<td>Orchestration ROI computes savings resulting from automated tasks and provides reports based on predicted and actual savings.</td>
<td>inactive</td>
<td></td>
<td>Performance Analytics, Cost Management</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Orchestration - ROI Premium</td>
<td>com.snc.runbook_snc</td>
<td>Orchestration - ROI premium reports provide detailed views of automation savings. These reports are available with activation of the Performance Analytics - Premium plugin.</td>
<td>active</td>
<td></td>
<td>Performance Analytics - Premium, Orchestration - ROI</td>
</tr>
<tr>
<td>Orchestration - SFTP</td>
<td>com.snc.orchestration</td>
<td>This is a simple activity pack that includes activities for common file transfer tasks and a complex workflow ready to use in real world applications.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration - SSH</td>
<td>com.snc.orchestration</td>
<td>This is an activity pack for SSH.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration - VMware Support</td>
<td>com.snc.runbook_snc</td>
<td>This is a VMware orchestration support. Includes a service catalog entry for ordering a virtual server with a choice of operating systems and performance/capacity class, and also configurations for Windows, Linux, and network information.</td>
<td>inactive</td>
<td></td>
<td>com.snc.orchestration.vmware, com.snc.runbook_automation</td>
</tr>
<tr>
<td>Orchestration - System Center Configuration Manager</td>
<td>com.snc.orchestration</td>
<td>This is an activity pack for client software distribution activities.</td>
<td>inactive</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Orchestration Activities - Chef</td>
<td>com.snc.orchestration.chef</td>
<td>Uses workflow technology to utilize activities that work with Opscode's Chef product. Please consult with your ServiceNow representative.</td>
<td>active</td>
<td></td>
<td>Orchestration, Web Service Consumer</td>
</tr>
<tr>
<td>Orchestration Activities - EC2</td>
<td>com.snc.orchestration.ec2</td>
<td>Uses workflow technology to issue Amazon EC2 activities that will execute the commands on Amazon cloud. Available by request</td>
<td>active</td>
<td></td>
<td>Amazon Web Services Core Components, Orchestration</td>
</tr>
<tr>
<td>Orchestration Activities - Puppet</td>
<td>com.snc.orchestration.puppet</td>
<td>Uses workflow technology to utilize activities that work with Puppet. Please consult with your ServiceNow representative.</td>
<td>active</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Orchestration Activities - VMware</td>
<td>com.snc.orchestration.vmware</td>
<td>Uses workflow, VMware and MID server technologies to allow VMware activities to execute commands on equipment inside the customer’s enterprise network. Available as a separate subscription</td>
<td>active</td>
<td></td>
<td>Orchestration</td>
</tr>
<tr>
<td>Ordered Email Processing</td>
<td>com.glide.email_ordered</td>
<td>Allows inbound email actions to be ordered, and programmatically stop processing.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Organization Management</td>
<td>com.snc.organization_management</td>
<td></td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Outbound Email Notifications</td>
<td>com.glide.email_outbound_email_notifications</td>
<td>Enables outbound email notifications.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Overview of cluster state</td>
<td>com.glide.system_status</td>
<td>Available by request and supporting business rules to allow single point overview of cluster state.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Overview Pages</td>
<td>com.glide.ui.overview</td>
<td>Framework for Overview pages.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.angular</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packages Call Removal Tool</td>
<td>com.glide.script_packages_call removal</td>
<td>Scans scripts for Packages calls to ServiceNow Java classes, proposes changes to replace them with alternate scriptable names, and facilitates the script changes. Packages calls to ServiceNow Java classes will eventually be disallowed in a future ServiceNow release, and this utility helps prepare an instance for that.</td>
<td></td>
<td>false</td>
<td>com.glide.script.packages_call_removal</td>
</tr>
<tr>
<td>Password Reset</td>
<td>com.glideapp.password_reset</td>
<td>Provides the ability to create self-service and service desk password reset processes for a ServiceNow instance.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.process.flow_formatter, com.glide.notification, com.glide.usageanalytics</td>
</tr>
<tr>
<td>Password Reset - Orchestration Add-on</td>
<td>com.glideapp.password.reset.add.on</td>
<td>Password reset add-on for enabling the use of ServiceNow orchestration. Includes support for Active Directory and remote SOAP based credentialed systems.</td>
<td>inactive</td>
<td>true</td>
<td>Password Reset, Orchestration</td>
</tr>
</tbody>
</table>

Available by request
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDF Generator</td>
<td>com.snc.pdf_generator</td>
<td>Provides a tool to generate PDF documents. Uses variables to pull information from a table into the document. Used to generate employment verification letters in the Human Resources application.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Analytics</td>
<td>com.snc.pa</td>
<td>Enables users to define and track key performance indicators (KPIs) and visualize these in scorecards and dashboards. Users can report and compare multiple time series, do advanced trend analysis, and compare their performance with preset targets.</td>
<td>active</td>
<td>false</td>
<td>com.snc.pa.dc, com.glide.ui.angular, com.glideapp.report, com.glideapp.home.publisher.</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Performance Analytics - Configuration Generator</td>
<td>com.snc.pa.configurationgenerator</td>
<td>Provides a configuration generator for creating a set of Performance Analytics indicators, breakdowns, dashboards, and widgets based on the task-derived table.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Content Pack - Change Management</td>
<td>com.snc.pa.change</td>
<td>Provides Performance Analytics content for change management. Installed with Performance Analytics - Premium.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Content Pack - Customer Service</td>
<td>com.sn_customerservice_pa</td>
<td>Provides Performance Analytics content for Customer Service. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.sn_customerservice, com.snc.pa.premium</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| Performance Analytics - Content Pack - Field Service Management  
**New in Geneva** | com.snc.work_management_pa | Provides Performance Analytics content for Field Service Management. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details. | inactive | true | com.snc.work_management, com.snc.pa.premium |
| Performance Analytics - Content Pack - Human Resources  
**New in Geneva** | com.snc.pa.hr_core | Provides Performance Analytics content for Human Resources core out-of-the-box KPIs. Activation of this plugin on production instances may require a separate Performance Analytics license. Contact ServiceNow for details. | inactive | false | com.snc.pa, com.snc.hr.core |
<p>| Performance Analytics - Content Pack - Incident SLA Management | com.snc.pa.sla | Provides Performance Analytics content for Incident SLA Management. Installed with Performance Analytics - Premium. | inactive | false | com.snc.pa |</p>
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Analytics - Content Pack - Problem Management</td>
<td>com.snc.pa.prob</td>
<td>Provides Performance Analytics content for Problem Management. Installed with Performance Analytics - Premium.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Content Pack - Request Management (Requested Item)</td>
<td>com.snc.pa.req</td>
<td>Performance Analytics content for Request Management. Installed with Performance Analytics - Premium.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Content Pack - Request Management (Requests)</td>
<td>com.snc.pa.req</td>
<td>Performance Analytics content for Request Management. out-of-the-box KPIs for Requests. Activation of this plugin on production instances may require a separate Performance Analytics license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Data Collector</td>
<td>com.snc.pa.dc</td>
<td>Performance Analytics - Data Collector. Installed with Performance Analytics.</td>
<td>active</td>
<td>false</td>
<td>com.snc.core.automation, com.glide.usageanalytics</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Performance Analytics - Example - LinkedIn</td>
<td>com.snc.pa.linkedin.feature_set</td>
<td>Automatically imports LinkedIn data historically and daily.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Example - Stocks Quotes</td>
<td>com.snc.pa.stock.feature_set</td>
<td>Automatically imports stock quotes data historically and daily.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Example - Twitter</td>
<td>com.snc.pa.twitter.feature_set</td>
<td>Automatically import Twitter data historically and daily.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Example - Context Sensitive Analytics for Incident</td>
<td>com.snc.pa.incident.context_sensitive_analytics.feature_set</td>
<td>Performance Analytics example adding ability to open PA context-sensitive PA dashboards in incident forms based on UI actions.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>Performance Analytics - Premium</td>
<td>com.snc.pa.premium</td>
<td>Installing this plugin will remove the limitations of the out of box Performance Analytics.</td>
<td>inactive</td>
<td>false</td>
<td>Performance Analytics</td>
</tr>
<tr>
<td>Personal Task Management</td>
<td>com.glide.ui.m.pt</td>
<td>Allows users to create and manage personal tasks.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.ui.m</td>
</tr>
<tr>
<td>Platform as a Service</td>
<td>com.snc.paas.application</td>
<td>Allows the development of custom applications to meet business needs.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.workflow</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Problem Management</td>
<td>com.snc.problem</td>
<td>Helps to identify the cause of an error in the IT infrastructure that is usually reported as occurrences of related incidents.</td>
<td>active</td>
<td>true</td>
<td>com.snc.service</td>
</tr>
<tr>
<td>Problem Tasks</td>
<td>com.snc.problem_task</td>
<td>Adds a Problem Task table with a reference to the Problem table.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Process Flow Formatter</td>
<td>com.snc.process_flow_formatter</td>
<td>Quickly summarizes multiple pieces of information about a process and displays the stages graphically at the top of a form.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Procurement</td>
<td>com.snc.procurement</td>
<td>Allows users to create purchase orders and obtain items for fulfilling service catalog requests.</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Project Management</td>
<td>com.snc.project_management_v3</td>
<td>Third generation suite of tools that aids in planning, organizing, and managing projects by applying basic task management processes.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.project_management_v3, com.snc.planned_task_v2, com.snc.time_card, com.snc.skills_management, com.snc.process_flow_formatter, com.snc.cost_management, com.snc.project_management_db_views, com.snc.project_workbench, com.snc.timeline_visualization, com.glide.ui.checklist</td>
</tr>
<tr>
<td>Project Management Import</td>
<td>com.snc.project_management_v2</td>
<td>An addition to the Project Management v2 pluging that makes importing MPP (Microsoft Project Files) possible.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.project_management_v2, Project Management v2</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Management TeamSpace 1</td>
<td>com.snc.ppm_teamspace_1</td>
<td>application</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.project_portfolio_suite</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a Project TeamSpace so that each team or each department like Marketing, Finance, IT-Team1, IT-Team2 can implement PPM suitable to their needs without overstepping each other.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Management TeamSpace 2</td>
<td>com.snc.ppm_teamspace_2</td>
<td>application</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.project_portfolio_suite</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a Project TeamSpace so that each team or each department like Marketing, Finance, IT-Team1, IT-Team2 can implement PPM suitable to their needs without overstepping each other.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Management TeamSpace 3</td>
<td>com.snc.ppm_teamspace_3</td>
<td>application</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.project_portfolio_suite</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a Project TeamSpace so that each team or each department like Marketing, Finance, IT-Team1, IT-Team2 can implement PPM suitable to their needs without overstepping each other.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Project Management TeamSpace 4</td>
<td>com.snc.ppm_teamspace_4_teamspace_4</td>
<td>a Project TeamSpace so that each team or each department like Marketing, Finance, IT-Team1, IT-Team2 can implement PPM suitable to their needs without overstepping each other.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.project_portfolio_suite</td>
</tr>
<tr>
<td>Project Management TeamSpace 5</td>
<td>com.snc.ppm_teamspace_5_teamspace_5</td>
<td>a Project TeamSpace so that each team or each department like Marketing, Finance, IT-Team1, IT-Team2 can implement PPM suitable to their needs without overstepping each other.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.project_portfolio_suite</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>----------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Protocol Profile Manager</td>
<td>com.glide.protocol_profile</td>
<td>Defines properties associated to protocols such as default port and keystore.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Puppet Configuration Management</td>
<td>com.snc.puppet</td>
<td>Defines properties associated to protocols such as default port and keystore.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Quiz Designer</td>
<td>com.glide.quiz_designer</td>
<td>Provides the ability to send scored questionnaires quickly and easily to one or more users. A quiz may have categories of questions that are assigned only to some users.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.assessment_designer, com.snc.assessment_core</td>
</tr>
<tr>
<td>Read-Only User Role</td>
<td>com.snc.read_only</td>
<td>Enable Read-Only user role functionality.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Relationship Layout New in Geneva</td>
<td>com.glide.ui.relationship_layout</td>
<td>Enable scoped relationships to be associated to out-of-scope related list views.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Release Management</td>
<td>com.snc.release_management</td>
<td>The Release Management v2 plugin is a rewrite of the original release management module. All products, releases, features, and release tasks are now planned_task extensions, and much of the project management functionality (Gantt charts, timelines, time cards) is now shared with release.</td>
<td>active</td>
<td>true</td>
<td>com.snc.release_management, com.snc.planned_task, com.snc.process_flow_formatter</td>
</tr>
<tr>
<td>Report - PDF Page Header Footer Templates</td>
<td>com.glideapp.report_page_hdrftr</td>
<td>Allows users to create templates that define the page header and footer layout for PDF files exported from ServiceNow.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.report, com.glideapp.report2</td>
</tr>
<tr>
<td>Report Charting v2</td>
<td>com.glideapp.report_charting_v2</td>
<td>Installs V2 of ServiceNow charts with HighCharts.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.report, com.glideapp.report2</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Report Engine - use summary table for reports</td>
<td>com.glideapp.report_engine</td>
<td>Causes the data from all reports, custom and standard, to be stored in the sys_report_summary table and separates the data from the rendering process for all reports. Report data is periodically purged from the sys_report_summary table (approximately every two hours).</td>
<td>false</td>
<td></td>
<td>com.glideapp.report2</td>
</tr>
<tr>
<td>Reporting Statistics Reports</td>
<td>com.glideapp.report_stat</td>
<td>Provides reports and dashboards on reporting statistics.</td>
<td>inactive</td>
<td>false</td>
<td>com.glideapp.report</td>
</tr>
<tr>
<td>Required Form Fields</td>
<td>com.snc.required_form</td>
<td>Allows the administrator to specify required fields that cannot be removed from a form.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Resource Management</td>
<td>com.snc.resource_management</td>
<td>Enables resource requesters and resource managers to plan, organize, and manage resources for both planned and unexpected work. Activating Resource Management automatically activates the Project Management plugin if it is not already active.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.schedule_loader, com.snc.process_flow_formatter, com.snc.cost_management</td>
</tr>
<tr>
<td>Resource Matching Engine</td>
<td>com.snc.matching_rule</td>
<td>Provides a Resource Matching Engine.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.skills_management</td>
</tr>
<tr>
<td><strong>New in Geneva</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REST API Provider</td>
<td>com.glide.rest</td>
<td>Provides a REST API framework to support RESTful services.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.angular, com.glide.ui.heisenberg</td>
</tr>
<tr>
<td>Restore Deleted Records</td>
<td>com.snc.undelete</td>
<td>Restores deleted records from audited tables and references to those records. Also restores any records that were deleted as a result of a cascade delete rule.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Delegation</td>
<td>com.snc.role_delegation</td>
<td>Allows an administrator to designate role delegators, who can delegate any role they have to members of their group.</td>
<td>active</td>
<td>false</td>
<td>com.glideapp.workflow</td>
</tr>
<tr>
<td>Runbook Automation - Amazon EC2 Integration for Service-Now</td>
<td>com.snc.ec2_snc_application</td>
<td>ServiceNow specific implementation of the Amazon EC2 training partner application. It provides a custom implementation of the Update DNS workflow.</td>
<td>inactive</td>
<td></td>
<td>Orchestration - Amazon EC2 Integration for Training Partners</td>
</tr>
<tr>
<td>Sales Force Automation application template</td>
<td>com.snc.sfa2_application</td>
<td>Provides tools to manage sales and marketing operations throughout the sales life cycle from lead generation through contract completion.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.process_flow_formatter</td>
</tr>
<tr>
<td>Schema Map Plugin</td>
<td>com.glideapp.schema_map</td>
<td>Enables a dynamic visual map feature for the CMDB.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Schema Map v3</td>
<td>com.glide.erd</td>
<td>Displays the details of tables and their relationships in a visual manner, allowing administrators to view and easily access different parts of the database schema.</td>
<td>active</td>
<td>false</td>
<td>com.glide.diagrammer</td>
</tr>
<tr>
<td>Script Templates</td>
<td>com.glide.script.templates</td>
<td>Provides templates for some script fields.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td><strong>New in Geneva</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Script Whitelist Manager</td>
<td>com.glide.script.whitelist</td>
<td>Provides temporary support for continued direct invocation of whitelisted Java Packages, Constructors, and Methods.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td><strong>New in Geneva</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scripted REST APIs</td>
<td>com.glide.scripted_rest_services</td>
<td>Provides a framework for building Scripted REST APIs.</td>
<td>active</td>
<td>false</td>
<td>com.glide.rest, com.glide.scripted_rest_services, com.glide.scripted_rest_services.errors</td>
</tr>
<tr>
<td><strong>New in Geneva</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scripted REST APIs - Error types</td>
<td>com.glide.scripted_rest_services.errors</td>
<td>A plugin component to Scripted REST APIs.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Scripted REST APIs - Internal</td>
<td>com.glide.scripted_rest_services.errors</td>
<td>A plugin component to Scripted REST APIs.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>SDLC - SCRUM</td>
<td>com.snc.sdlicapplication</td>
<td>Adds a release process specific to a Scrum development process. Sits on top of the SDLC application, adding additional agile notions like epics, stories, and sprints.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.sdlic</td>
</tr>
<tr>
<td>SDLC - Scrum Process Pack</td>
<td>com.snc.sdlicapplication</td>
<td>Wraps a formal process over the core SDLC-SCRUM offering. If you already have a customized version of SDLC-SCRUM, please refer to the discussion of deleting previous customizations before activating this plugin.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.model, com.snc.sdlic, com.glideapp.custom_charts, com.glide.sorting, com.glide.ui.ng</td>
</tr>
<tr>
<td>Secondary Database Pools</td>
<td>com.glide.secondary_db_pools</td>
<td>Define secondary database pools and group them into categories.</td>
<td>inactive</td>
<td>false</td>
<td>Progress Tracker</td>
</tr>
<tr>
<td>Security Incident Response</td>
<td>com.snc.security_incident</td>
<td>Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_management, com.snc.governance_core, com.snc.task_outage, com.snc.treemap, com.snc.security_support.sir</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Security Incident Response Event</td>
<td>com.snc.security_incident</td>
<td>Activates this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.security_incident, com.glideapp.itom.snac</td>
</tr>
<tr>
<td>Event Management support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>New in Geneva</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Jump Start (ACL Rules)</td>
<td>com.snc.system_security</td>
<td>Adds ACL rules to provide a jump start on securing many system tables, making it easier for an organization to get into production more quickly.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Self Service Password Reset</td>
<td>com.snc.password_reset</td>
<td>Allows locally authenticated users to request a temporary password if they forget their current one.</td>
<td>inactive</td>
<td>false</td>
<td>com.glideapp.password_reset</td>
</tr>
<tr>
<td>Sendmail Support</td>
<td>com.glide.sendmail</td>
<td>Supports forwarding of emails from *nix sendmail using SOAP into ECC queue.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td><strong>Available by request</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server-side JavaScript Debugger</td>
<td>com.glide.debugger</td>
<td>Allows application developers and administrators to efficiently debug scripts that drive the applications they develop and support.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.angular, com.glide.ui.zepto, com.glide.ui.font_icons</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Side Monitoring for instance checkin</td>
<td>com.glide.instance_monitor</td>
<td>Server side parsing and storage for instance stat flow back to server.</td>
<td>inactive</td>
<td>false</td>
<td>Glide Relational Implementation of an RRDB</td>
</tr>
<tr>
<td>Server side system load test utility</td>
<td>com.glide.load_test</td>
<td>Provides simulated, repeatable load on an application server. Simulates UI rendering, updates, and other activities.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Server side system load test utility demo data</td>
<td>com.glide.load_test.demo</td>
<td>Demo data (and sample tests) for load test plugin.</td>
<td>inactive</td>
<td>false</td>
<td>Server side system load test utility</td>
</tr>
<tr>
<td>Service Analytics New in Geneva</td>
<td>com.snc.sa.analytics</td>
<td>Service Analytics.</td>
<td>inactive</td>
<td></td>
<td>Event Management</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Service Catalog core applications</td>
<td>com.glideapp.servicecatalog</td>
<td>Allows customers to order predefined, bundled goods and services from your IT organization or other departments.</td>
<td>active</td>
<td>true</td>
<td>com.glideapp.servicecatalog, com.glideapp.servicecatalog.execution_plan, com.glideapp.servicecatalog.item_designer, com.glide.currency</td>
</tr>
<tr>
<td>Service Catalog Scoped API</td>
<td>com.glideapp.servicecatalog.scoped.api</td>
<td>Service Catalog Scoped API to support application creation on Service Catalog platform.</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Service Creator</td>
<td>com.glide.service_creator</td>
<td>Enables a department to offer custom services through the service catalog, such as the HR department offering tuition reimbursement for further education.</td>
<td>active</td>
<td>true</td>
<td>com.glide.ui.ng.cc, com.glideapp.servicecatalog.service_creator</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Service Desk Call</td>
<td>com.snc.service_desk</td>
<td>Provides a great Service Desk application that enables service desk staff to collect information in a call that does not relate to a specific process. The call will be transferred to an incident, problem, change request or service catalog request. Replaces the Best Practice - New Call Wizard plugin and the New Ticket Module.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Service level management</td>
<td>com.snc.sla</td>
<td>Provides a greatly enhanced version of the default SLA engine.</td>
<td>active</td>
<td>true</td>
<td>com.glideapp.workflow, com.glide.schedules, com.glide.relative_duration</td>
</tr>
<tr>
<td>Service Management Core</td>
<td>com.snc.service_management.core</td>
<td>Installs the core Service Management items used to allow other service-related plugins to work, such as Field Service, Facilities, HR, Legal, Finance, Marketing and the custom app creator.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.automatic_assignment, com.snc.asset_management, com.snc.process_flow_formatter, com.snc.state_flows, com.snc.knowledge3, com.snc.skills_management, com.snc.territory_management, com.snc.document_management, com.snc.task_activity, com.snc.service_management, com.glide.encryption, com.snc.common_workbench, com.glide.ui.checklist</td>
</tr>
<tr>
<td>Service Management Geolocation</td>
<td>com.snc.service_management.geolocation</td>
<td>Provides Service Management geolocation capabilities.</td>
<td>active</td>
<td>true</td>
<td>com.snc.geolocation</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Plugin</th>
<th>ID</th>
<th>Description</th>
<th>Active</th>
<th>Has Demo Data?</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Management Geolocation Mobile</td>
<td>com.snc.service_management_m</td>
<td>Adds a menu in the new mobile UI for Service Management Geolocation.</td>
<td>inactive</td>
<td>false</td>
<td>com.glide.ui.m, com.snc.service_management_geolocation</td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Mapping</td>
<td>com.snc.service_mapping</td>
<td>Top Down Service Discovery.</td>
<td>inactive</td>
<td></td>
<td>Event Management and Service Mapping Core</td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Available as a separate subscription</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Modeling</td>
<td>com.snc.cmdb.service_modeling</td>
<td>Service Modeling infrastructure for Service Modeling used in Service Mapping and Delivery.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.cmdb.enterprise</td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Portfolio Management</td>
<td>com.snc.service_portfolio</td>
<td>Application organization to document the business services it provides using a standardized, structured format. Performance against availability commitments is calculated and can be displayed in a homepage.</td>
<td>inactive</td>
<td>true</td>
<td>com.glideapp.servicecatalog, com.glideapp.summary_report_engine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Service Portfolio Management - SLA Commitments</td>
<td>com.snc.service_portfolio.sla</td>
<td>Allows commitments to be defined by an SLA, so that staff can track how efficiently the service desk meets commitments for a service offering.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_portfolio, com.snc.sla</td>
</tr>
<tr>
<td>ServiceNow Configuration Automation</td>
<td>com.snc.snc_config.application</td>
<td>Provides support for the management of ServiceNow configuration management automation. Please consult with your ServiceNow representative.</td>
<td>inactive</td>
<td></td>
<td>Orchestration, Configuration Automation</td>
</tr>
<tr>
<td>ServiceNow Edge Encryption New in Geneva Available as a separate subscription</td>
<td>com.glide.edgeencryption</td>
<td>Enables ServiceNow Edge Encryption.</td>
<td>inactive</td>
<td></td>
<td>ServiceNow Edge Encryption Core</td>
</tr>
<tr>
<td>ServiceNow Search Trends Available by request</td>
<td>com.glide.service_now_search_trends</td>
<td>Enables sharing of Search Trends between ServiceNow Instances.</td>
<td>inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SessionDebug New in Geneva</td>
<td>com.glide.sessiondebug</td>
<td>Provides SessionDebug statements and filtering.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Skills Management</td>
<td>com.snc.skills_management</td>
<td>Enables an administrator to assign configured competencies, called skills, to groups or individual users. These skills can then be used to determine which users and groups can be assigned particular tasks.</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>SLA Contract Add-on</td>
<td>com.snc.sla.contract</td>
<td>Extends the existing SLA functionality by utilizing a contract as the master document that houses all appropriate data needed to drive task SLA processing.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.sla</td>
</tr>
<tr>
<td>SLA Timeline</td>
<td>com.snc.sla.timeline</td>
<td>Provides the ability to view an SLA in a timeline.</td>
<td>active</td>
<td>false</td>
<td>com.snc.sla</td>
</tr>
<tr>
<td>SM Planned Maintenance</td>
<td>com.snc.planned_maintenance</td>
<td>Allows setup and configuration for repeating and triggered requests. Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_management</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SNC Access Control</td>
<td>com.snc.snc_access</td>
<td>Allow control customers to control which SNC employees may access their instance, and when</td>
<td>inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Available by request</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNC Code Editor</td>
<td>com.glide.snc_code</td>
<td></td>
<td>active</td>
<td>false</td>
<td>com.snc.knowledge3, com.glide.ui.ng.amb</td>
</tr>
<tr>
<td>ServiceNow Configuration Automation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social Q&amp;A</td>
<td>com.sn_kb_social</td>
<td>Allow users to ask and respond to questions and to vote on questions and answers. Social Q&amp;A extends the Knowledge application and uses existing Knowledge functionality such as user criteria and multiple knowledge bases.</td>
<td>active</td>
<td>true</td>
<td>com.snc.knowledge3, com.glide.ui.ng.amb</td>
</tr>
<tr>
<td>New in Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Asset Management</td>
<td>com.snc.software</td>
<td>Systematically tracks, evaluates, and manages software licenses and software usage.</td>
<td>active</td>
<td>true</td>
<td>com.snc.sam.core</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>iD</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Software Asset Management - IBM PVU Process Pack</td>
<td>com.snc.sam.ibmpvu.pp</td>
<td>Add-on to the Software Asset Management plugin that provides the capability to manage software that is licensed under the IBM Processor Value Units licensing model.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.software_asset_management</td>
</tr>
<tr>
<td>Software Asset Management - Oracle Process Pack</td>
<td>com.snc.sam.oracle.pp</td>
<td>Add-on to the Software Asset Management plugin that provides the capability to manage software that is licensed under Oracle licensing model.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.software_asset_management</td>
</tr>
<tr>
<td>Software Asset Management Core</td>
<td>com.snc.sam.core</td>
<td>Provides the base tables for software asset management. Includes software installations, usages, suite calculations, and discovery models.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Software Asset Management Extensions</td>
<td>com.snc.sam</td>
<td>Provides Software asset management improvements, including license controls.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Software Development Lifecycle (SDLC)</td>
<td>com.snc.sdlc application</td>
<td>Extends the Release Management v2 plugin by adding some new structures to accommodate the software development life cycle. This plugin is designed to accommodate most non-agile development methodologies, including the common waterfall method of development.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.release_management</td>
</tr>
<tr>
<td>SSO provided by Okta, Inc.</td>
<td>com.snc.sso.okta</td>
<td>Provides single sign-on access to ServiceNow instances through Okta.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.integration.sso.saml20.update1</td>
</tr>
<tr>
<td>Staging tables</td>
<td>com.glideapp.staged_tables</td>
<td>Manages staging tables.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>State Flows</td>
<td>com.snc.state_flows</td>
<td>Enables advanced users to customize the state flow of any task table that uses states. This plugin is activated when Work_Management (in versions prior to Fuji) is activated and contains records that define the default work management state flow.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Stats Tools</td>
<td>com.glide.db.query_stats</td>
<td>Records statistics for system activities that affect performance.</td>
<td>active</td>
<td>false</td>
<td>com.glide.monitor.round_robin_database</td>
</tr>
<tr>
<td><strong>New name in Geneva; formerly Query Stats</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Streamline applications for Mid-market sized accounts</td>
<td>com.snc.boot_profile</td>
<td>Streamline applications for mid-market sized accounts - turn off non-core Applications and trim Modules.</td>
<td>inactive</td>
<td>false</td>
<td>Boot profile support routines</td>
</tr>
<tr>
<td><strong>Available by request</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Studio</td>
<td>com.glide.dev-studio</td>
<td>Studio allows developers to add and update application files.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td><strong>New in Geneva</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subscription Based Notifications</td>
<td>com.glide.notification</td>
<td>Allows users to subscribe to notifications on a task or CI without being on the watchlist or being one of the assigned users.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui_policy, com.glide.email_notification_preview</td>
</tr>
<tr>
<td>Survey Designer</td>
<td>com.glide.survey_designer</td>
<td>Survey designer.</td>
<td>active</td>
<td>false</td>
<td>com.glide.assessment_designer</td>
</tr>
<tr>
<td><strong>Available by request</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax Editor</td>
<td>com.glide.syntax_editor</td>
<td></td>
<td>active</td>
<td>false</td>
<td>com.glide.snc_code_editor</td>
</tr>
<tr>
<td>System Applications Core</td>
<td>com.snc.apps_application</td>
<td>Core applications development.</td>
<td>active</td>
<td>false</td>
<td>com.snc.apps_picker</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>System Applications Support</td>
<td>com.snc.apps_complete</td>
<td>Provides support for creating and managing applications.</td>
<td>active</td>
<td>false</td>
<td>com.snc.apps, com.snc.apps_picker, com.snc.apps_access, ...</td>
</tr>
<tr>
<td>System Import Sets</td>
<td>com.glide.system_import</td>
<td>Provides the functionality for import sets.</td>
<td>active</td>
<td>false</td>
<td>com.snc.automation, com.glide.ui_policy, com.glide.system_import_data_source</td>
</tr>
<tr>
<td>System Import Sets - Tests</td>
<td>com.glide.system_import</td>
<td>Test plugin sets. Test plugin sets. Contains table structures, maps, and applications for a convenient testing environment</td>
<td>inactive</td>
<td>false</td>
<td>System Import Sets</td>
</tr>
<tr>
<td>System Update Set Picker</td>
<td>com.glide.system_update</td>
<td>Allows users to choose an update set for tracking customizations.</td>
<td>active</td>
<td>false</td>
<td>com.glide.system_update_set_picker</td>
</tr>
<tr>
<td>System Update Sets (viewer)</td>
<td>com.glide.local_update</td>
<td>Supports viewing contents of update sets.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>System Update Sets (with remote update set support)</td>
<td>com.glide.system_update</td>
<td>Facilitates moving customizations between systems. Supports viewing contents of update sets.</td>
<td>active</td>
<td>false</td>
<td>com.snc.apps_hub, com.glide.system_update_set_picker, com.glide.system_update_set_preview</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>System Update Sets Preview</td>
<td>com.glide.system_update_set_preview</td>
<td>Allows users to preview the changes that will be performed by an update set and predict whether there will be any collisions in attempting to apply the update set.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>System Web Services</td>
<td>com.glide.web_service_application</td>
<td>Provides a series of web service import sets.</td>
<td>active</td>
<td>false</td>
<td>com.glide.web_service_provider, com.glide.web_service_import_sets, com.snc.web_service_import_set_tables</td>
</tr>
<tr>
<td>Table Hierarchy Conversion Available by request</td>
<td>com.snc.ha.clone.hierarchy</td>
<td>Flatten table hierarchies - eliminates joins on tables in the hierarchy.</td>
<td>inactive</td>
<td>false</td>
<td>High Availability Cloning</td>
</tr>
<tr>
<td>Tablet Device Support - iPad with iOS 6+</td>
<td>com.glide.ui.tablet</td>
<td>Provides a UI supporting nearly full-product functionality on the Apple iPad.</td>
<td>active</td>
<td>false</td>
<td>com.glide.context_help, com.glide.ui.font_icons, com.glide.ui.tablet.theme</td>
</tr>
<tr>
<td>Task Activities New in Geneva</td>
<td>com.snc.task_activity</td>
<td>Enables support for activities on task tables.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Task-Outage Relationship</td>
<td>com.snc.task_outage</td>
<td>Allows users to create an outage from an Incident and a Problem form. Incidents and problems have a many-to-many relationship with outages.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Team Development</td>
<td>com.snc.apps_hub.application</td>
<td>Supports parallel development on multiple, sub-production ServiceNow instances by providing branching operations, the ability to compare a development instance to other development instances, and a central dashboard for all team development activities.</td>
<td>active</td>
<td>false</td>
<td>com.snc.apps, com.glide.ui.angular</td>
</tr>
<tr>
<td>Territory Management</td>
<td>com.snc.territory_management</td>
<td>Implementation of geographical territory covered by users (Agent) or groups (Field Support Groups).</td>
<td>inactive</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Test Management</td>
<td>com.snc.test_mgmt</td>
<td>Provides a tool for manual software testing.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.planned_task_v2</td>
</tr>
<tr>
<td>Test Suite/Case Management</td>
<td>com.snc.test_suite</td>
<td>Test Suite: Plugin to allow Test definition and recording.</td>
<td>inactive</td>
<td>true</td>
<td>Context Ranking</td>
</tr>
<tr>
<td>Text Analytics</td>
<td>com.glide.ts_analytics</td>
<td>Provides analysis of text data in a body of records over time.</td>
<td>inactive</td>
<td>false</td>
<td>Zing Text Search</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Text Index Attachments</td>
<td>com.glide.text_index_attachments</td>
<td>indexes attachments so that attachments can appear in searches.</td>
<td>active</td>
<td>false</td>
<td>com.glide.text_index</td>
</tr>
<tr>
<td>Text Search</td>
<td>com.glide.text_search</td>
<td>Allows global text search.</td>
<td>active</td>
<td>false</td>
<td>com.glide.text_index</td>
</tr>
<tr>
<td>Time card management</td>
<td>com.snc.time_card_application</td>
<td>Works with the Task table to record time worked on projects, incidents, problems, and change requests. Task assignees can record time worked in the <strong>Time worked</strong> field on a task record or enter hours directly onto a time card.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Timeline Visualization</td>
<td>com.snc.timeline</td>
<td>Enables graphical representation of activities over time to provide a high-level view of strategic and operational activities in your organization such as incidents, problems, changes, and projects. An out-of-the-box visualization provided by this plugin is the CIO Roadmap. This roadmap shows projects grouped by portfolios. Organization leaders can use the CIO roadmap to monitor and evaluate the status of current and upcoming projects.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.project_management_v3</td>
</tr>
<tr>
<td>Tiny URL Support</td>
<td>com.glide.tiny_url</td>
<td>Automatically generates shortened URLs that prevent page rendering errors in Microsoft Internet Explorer caused by long URLs.</td>
<td>active</td>
<td>false</td>
<td>com.snc.tiny_url</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>TinyMCE HTML Field Editor</td>
<td>com.glide.editor.tinymce</td>
<td>Enables users to edit HTML fields with the TinyMCE editor instead of the legacy (htmlArea) editor.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Transaction Design Scope</td>
<td>com.glide.transaction_scope</td>
<td>Handles transaction design scope management.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Transaction Quotas</td>
<td>com.glide.quota</td>
<td>Allows definition of quota policies for different types of transactions. A transaction quota cancels any transaction in violation of the policy and notifies the user of the cancellation.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Tree map</td>
<td>com.snc.treemap</td>
<td>Enables support for treemap view on any applications.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.pa</td>
</tr>
<tr>
<td>UI Components</td>
<td>com.glideapp.ui_components</td>
<td>Provides all common angular components for apps.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.ng, com.glide.ui.angularui, com.glide.ui.heisenberg</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>UI11 - Navpage layout with multiple panes</td>
<td>com.glide.ui11</td>
<td>Enables UI11, a user interface that organizes the UI layout in multiple resizable panes, bookmarks, and flyouts.</td>
<td>active</td>
<td>false</td>
<td>com.glide.db_images</td>
</tr>
<tr>
<td>UI16 New in Geneva</td>
<td>com.glide.ui.ui16</td>
<td>Enables UI16, a user interface that provides an updated look and usability improvements. Notable features include real-time form updates, user presence, a redesigned application navigator with tabs for favorites and history, and enhanced activity streams.</td>
<td>active</td>
<td>false</td>
<td>com.glide.ui.doctype, com.glide.ui.concourse, com.glide.ui.form_presence, com.glide.ui.snippets</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User Guide</td>
<td>com.glide.user_guide</td>
<td>Provides the ability to create end user help documentation that is specific to the policies and procedures of your organization. A default help page is provided in the base system that displays UI16 help documents for system navigation and other basic operations.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>User Registration Request</td>
<td>com.snc.user_registration.feature_set</td>
<td>Provides the ability for unregistered users to request access to a ServiceNow instance.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Vendor Performance</td>
<td>com.snc.vendor_performance</td>
<td>Provides capabilities to measure, manage, and track vendor data and compare performance characteristics in unique graphical views.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.assessment_core</td>
</tr>
<tr>
<td>Vendor Ticketing</td>
<td>com.snc.vendor_ticket</td>
<td>Provides expanded functionality to obtain vendor performance data by tracking vendor incidents and SLAs.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.service_portfolio, com.snc.service_portfolio.sla, com.snc.task_outage, com.snc.vendor_performance</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Version Management</td>
<td>com.snc.version.application</td>
<td>Provides the ability to track, compare, and revert to multiple versions of table records.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Version Support</td>
<td>com.glideapp.version</td>
<td>Supports tracking versions of files that are stored in update sets, including the ability to compare and revert to previous versions.</td>
<td>active</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Virtualization Configuration Automation Integration</td>
<td>com.snc.virtualization_config_auto.application</td>
<td>Integrates virtualization and configuration automation applications.</td>
<td>inactive</td>
<td>false</td>
<td>com.glideui.ng, com.glideui.font_icons, com.glide.task, com.glideui.checklist, com.glideui.amb</td>
</tr>
<tr>
<td>Visual Task Boards</td>
<td>com.glide.ui.vtb</td>
<td>Allows users to organize, modify, and track progress of multiple tasks from an intuitive, Kanban-inspired interface.</td>
<td>active</td>
<td>false</td>
<td>com.glideui.ng, com.glideui.font_icons, com.glide.task, com.glideui.checklist, com.glideui.amb</td>
</tr>
<tr>
<td>Vulnerability Response</td>
<td>com.snc.vulnerability</td>
<td>Activation of this plugin on production instances may require a separate license. Contact ServiceNow for details.</td>
<td>inactive</td>
<td>true</td>
<td>com.snc.sam.core, com.snc.security_support.vul</td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Web Service Consumer</td>
<td>com.glide.web_service_consumer</td>
<td>Provides a SOAP Message module for developing, prototyping, and saving outbound SOAP messages that can be reused in business rules and scripts.</td>
<td>active</td>
<td>false</td>
<td>com.glide.vars, com.glideapp.ecc</td>
</tr>
<tr>
<td>Web Service Import Set Tables</td>
<td>com.snc.web_service_import_set_tables</td>
<td>Direct web services and scripted web services in providing a web service interface to import set tables.</td>
<td>active</td>
<td>false</td>
<td>com.glide.web_service_import_sets, com.glide.web_service_application</td>
</tr>
<tr>
<td>Web Service Import Sets</td>
<td>com.glide.web_service_import_sets</td>
<td>Direct web services and scripted web services in providing a web service interface to import sets.</td>
<td>active</td>
<td>false</td>
<td>com.glide.table_editor, com.glide.web_service_provider</td>
</tr>
<tr>
<td>Web Service Provider - Common</td>
<td>com.glide.web_service_provider</td>
<td>Provides scripted web service and SOAP message resources.</td>
<td>active</td>
<td>false</td>
<td>com.glide.vars, com.glide.system_property_categories, com.glide.system_import_set</td>
</tr>
<tr>
<td>Web Service Provider - Custom WSDL</td>
<td>com.glide.static_wsdl</td>
<td>Provides the ability to create a scripted web service to accept any WSDL format.</td>
<td>inactive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Plugin</td>
<td>ID</td>
<td>Description</td>
<td>Active</td>
<td>Has Demo Data?</td>
<td>Dependencies</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Web Service Provider - Scripted</td>
<td>com.glide.custom_web_service</td>
<td>Enables users to create new web services that are not addressed by the system. Allows a user to define input and output parameters and use JavaScript to do everything in between.</td>
<td>active</td>
<td>false</td>
<td>com.glide.web_service_provider, com.glide.web_service_consumer</td>
</tr>
<tr>
<td>WebKit HTML to PDF</td>
<td>com.snc.whtp</td>
<td>Enables the instance to use the service WebKit HTML to PDF.</td>
<td>inactive</td>
<td>false</td>
<td>com.snc.platform.security.oauth</td>
</tr>
<tr>
<td>Workflow Authoring Tools</td>
<td>com.glideapp.workflow_authoring</td>
<td>Allows you to define and modify workflows by arranging and connecting activities with transitions.</td>
<td>active</td>
<td>true</td>
<td>com.glideapp.workflow</td>
</tr>
<tr>
<td>Workflow Runtime Engine</td>
<td>com.glideapp.workflow</td>
<td>Enables the creation of workflows that drive automated processes. This may entail generating tasks based on conditions, running scripts, generating approvals, or other actions. Satisfies the same need as the Execution Plans plugin but with greater control and an easier interface.</td>
<td>active</td>
<td>false</td>
<td>com.glide.diagrammer, com.glide.vars, com.glide.schedules, com.glide.relative_duration, com.glide.web_service_application, com.glide.service_api, com.snc.datastructure, com.glideapp.live_feed</td>
</tr>
</tbody>
</table>
Available system properties

This page describes properties available to an instance.

Some properties are available on a system properties form, but some lesser-used properties are only available from the System Property [sys_properties] table. In some cases, the property does not exist in a base instance, but can be added if you need to change the value.

Table 316: Available System Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.glide.attachment.max_size</td>
<td>Sets the maximum file attachment size in megabytes. Leave the field empty to allow attachments up to a maximum of 1GB.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1GB</td>
</tr>
<tr>
<td></td>
<td>• Location: [System Properties &gt; Security]</td>
</tr>
<tr>
<td>com.glide.csv.loader.ignore_non_parseable_lines</td>
<td>Enables (true) or disables (false) ignoring one or more lines (rows) in an import set that contain bad data, such as a row that is missing a column of data.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.glide.csv.loader.max_errors_allowed</td>
<td>Sets the maximum number of lines (rows) that an import can ignore before failing. If the import succeeds, the import lists the number of rows ignored due to errors.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 100</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.glide.email.max_read</td>
<td>Specifies the maximum number of emails a POP3 reader should process concurrently.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 20</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| com.glide.loader.verify_target_field_size | Controls whether import set fields can automatically increase in size during an import (true) or not (false). By default, data that exceeds the import field size is truncated. Set this property to true to allow any import set field to increase the column size to match the length of the data.  
  - Type: true | false  
  - Default value: false  
  - Location: add to the System Property [sys_properties] table |
| com.glide.ssl.read.timeout | Sets the time-out value in seconds for SSL connections during read operations. Typically, this setting is used as part of LDAPS. If you enter timeout values for both this system property and the LDAP Read timeout field, the lowest timeout value takes precedence.  
  - Type: integer  
  - Default value: 10  
  - Location: add to the System Property [sys_properties] table. |
| com.snc.apps.publish.maxrows | Defines the maximum number of data records to include when publishing an application.  
  - Type: integer  
  - Default value: 1,000  
  - Location: System Property [sys_properties] table |
| com.snc.iam.log_level | Logging level for the business rule MapUpstreamImpactedCI. Debug is the most detailed option with full trace of how the Impacted CI List is calculated. Error is the minimal logging option with only severe errors being logged.  
  - Type: string  
  - Default value: info  
  - Possible values: debug,info,error  
  - Location: System Property [sys_properties] table |
| com.snc.on_call_rotation.reminders.showtz | Specifies whether to show a user's time zone  
  - Type: true | false  
  - Default value: false  
  - Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.snc.pa.dc.max_breakdown_elements_limit</td>
<td>Maximum number of breakdown elements for a breakdown to be included in data collection.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 10000</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.snc.pa.dc.max_error_count</td>
<td>Maximum errors that may occur before data collection is stopped.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 500</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.snc.pa.dc.max_records</td>
<td>Maximum number of records that are stored during a data collection.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5000</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.snc.pa.dc.max_row_count_indicator_source</td>
<td>Maximum number of rows that are allowed to be fetched from an Indicator Source.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 50000</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.snc.pa.dc.script_timeout</td>
<td>Maximum time in seconds a script is allowed to run during a data collection cycle.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 30</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.snc.project.default_schedule</td>
<td>Stores the sys_ID if the default schedule attached to projects.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>com.snc.project.loglevel</td>
<td>Automatically resubmits timed-out Ajax requests.</td>
</tr>
<tr>
<td></td>
<td>• Type: choice list</td>
</tr>
<tr>
<td></td>
<td>• Default value: 0 (debugging disabled)</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.snc.project.wbs_gantt</td>
<td>Automatically resubmits timed-out Ajax requests.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: Project &gt; Administration &gt; Properties</td>
</tr>
<tr>
<td>com.snc.task.associate_ci</td>
<td>List of all the task types where user wants to associate CI's using a List system.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default change_request</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; All &gt;</td>
</tr>
<tr>
<td>com.snc.time_card.default_rate</td>
<td>(Cost Management plugin) Sets a default hourly rate to use if no labor rate cards apply to the user.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 50</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>com.snc.time_worked.update_task_timer</td>
<td>Enables (true) or disables (false) updating of the task timer value based on changes to the task time worked records. This is accomplished through the Update task timerbusiness rule.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>css.assessment.caption.background.color</td>
<td>Sets the background color of the caption on assessment and survey questionnaires.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: #eee</td>
</tr>
<tr>
<td></td>
<td>• Location: Assessments &gt; Admin &gt; Assessment Properties</td>
</tr>
<tr>
<td></td>
<td>• Survey Management &gt; Administration &gt; Properties</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>css.assessment.caption.font.color</td>
<td>Sets the font color of the caption text on assessment and survey questionnaires.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: #ffffff</td>
</tr>
<tr>
<td></td>
<td>• Location:</td>
</tr>
<tr>
<td></td>
<td>• Assessments &gt; Admin &gt; Assessment Properties</td>
</tr>
<tr>
<td></td>
<td>• Survey Management &gt; Administration &gt; Properties</td>
</tr>
<tr>
<td>css.assessment.question.header.background.color</td>
<td>Sets the background color of question headers on assessment and survey questionnaires.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: #767676</td>
</tr>
<tr>
<td></td>
<td>• Location:</td>
</tr>
<tr>
<td></td>
<td>• Assessments &gt; Admin &gt; Assessment Properties</td>
</tr>
<tr>
<td></td>
<td>• Survey Management &gt; Administration &gt; Properties</td>
</tr>
<tr>
<td>css.tablet.gradient.start</td>
<td>Start color of the gradient for the tablet UI header.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: #666</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Tablet UI Properties</td>
</tr>
<tr>
<td>css.tablet.gradient.end</td>
<td>End color of the gradient for the tablet UI header.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: #111</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Tablet UI Properties</td>
</tr>
<tr>
<td>css.tablet.headerfooter.text.color</td>
<td>Color of the text and icons in the tablet UI header and footer.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: lightgrey</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Tablet UI Properties</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.apps.hub.current</td>
<td>URL of the team development parent instance.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.attachment.extensions</td>
<td>Comma-separated list of file extensions that can be attached. No value means there are no restrictions.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: <a href="#">System Properties &gt; Security</a></td>
</tr>
<tr>
<td>glide.attachment.role</td>
<td>Comma-separated list of roles that can create attachments.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: public</td>
</tr>
<tr>
<td></td>
<td>• Location: <a href="#">System Properties &gt; Security</a></td>
</tr>
<tr>
<td>glide.authenticate.multisso.login_locate.user_field</td>
<td>Identifies a common login identifier.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: user_name</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.authenticate.sso.saml2.require_signed_authnrequest</td>
<td>Enables the Identity Provider’s Single sign on service to receive a signed AuthnRequest</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.banner.image.url</td>
<td>URL used when clicking the banner image.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: home.do</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.banner.image.url_target            | Target frame used when clicking the banner image. Use gsft_main for the main frame, _top to replace the current browser window, _blank for a new window or tab.  
  • Type: string  
  • Default value: gsft_main  
  • Location: System Property [sys_properties] table                                                                                                                                                                                                                                                                                                                                                       |
| glide.businessrule.async_condition_check | Specifies if the instance checks the condition statement of async business rules (the **When** field is set to async) prior to running the business rule. If this property is set to true, the instance evaluates the async business rule conditions a second time before running the rule. If this property is set to false or missing, the instance does not evaluate the conditions a second time.  
  • Type: true | false  
  • Default value: false  
  • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                                                                                           |
| glide.businessrule.callstack             | Determines whether business rule executions are logged in the daily log when they start and finish (true) or are not logged (false). Logging is useful for troubleshooting a problem where you need to know which business rules are running and in which order.  
  • Type: true | false  
  • Default value: false  
  • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                                                                                           |
| glide.chart.decimal.precision            | Controls the rounding precision of non-currency numeric values displayed on charts. This property has a maximum possible value of 4. Currency values always specify a precision of 2.  
  • Type: integer  
  • Default value: 2  
  • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                                                                                           |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.chat.invite_fields</td>
<td>(Chat plugin) Comma-separated list of fields (glide_list or references to sys_user or sys_group) used to generate the invitations when creating a chat room from a task. Users can select a check box for each of the specified fields to invite users referenced by the fields to the chat room.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: Social IT &gt; Chat Administration &gt; Properties</td>
</tr>
<tr>
<td>glide.chat_room.create_roles</td>
<td>(Chat plugin) Comma-separated list of roles that are allowed to create chat rooms.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: Social IT &gt; Chat Administration &gt; Properties</td>
</tr>
<tr>
<td>glide.chat.show_emoticons</td>
<td>(Chat plugin) Enables or disables rendering text emoticons as images.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: Social IT &gt; Chat Administration &gt; Properties</td>
</tr>
<tr>
<td>glide.cmdb_model.display_name.shorten</td>
<td>When set to true, generates shorter display names for product models if the name of the product model contains the manufacturer name.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.cookies.http_only</td>
<td>Enables (true) or disables (false) the generation of HTTP only cookies. Set this property to false to use Approval with E-Signature.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.cost_mgmt.calc_actual_cost</td>
<td>(Cost Management plugin) Sums all task expense lines and adds the total to the <strong>Work cost</strong> field on the task record when an expense line is created for any task with a <strong>Type</strong> of <strong>Planned task</strong>.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.cost_mgmt.debug</td>
<td>(Cost Management plugin) Enables debugging of cost management processing. All logging events are recorded in the Financial Management Log [fm_log] table. This feature should only be enabled during initial testing or when troubleshooting because it can generate a large number of log records.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.cost_mgmt.process_task_cis</td>
<td>(Cost Management plugin) Creates expense lines to affected configuration items when creating a task expense line.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.cost_mgmt.service_allocation.method</td>
<td>(Cost Management plugin) Defines whether business service to cost center allocation costs should be calculated based on total units or allocated units.</td>
</tr>
<tr>
<td></td>
<td>• Type: choice list</td>
</tr>
<tr>
<td></td>
<td>• Default value: all_units</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.csv.export.line_break</td>
<td>Enables the user to control how exported CSV data appears in Notepad. Valid values are LF for a line feed between records and CRLF for a carriage return followed by a line feed.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: LF</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.custom.ip.authenticate.allow</td>
<td>Comma-separated list or range of IP addresses that are allowed access to view the stats.do, threads.do, and replication.do pages.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table.</td>
</tr>
<tr>
<td>glide.db.clone.allow_clone_target</td>
<td>Enables (true) or disables (false) use of a sub-production instance as the target for a system clone.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false for production instances, true for sub-production instances</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.db.large.threshold</td>
<td>Sets the number of rows above which a table is considered large and uses a different method of querying for results. When this property is absent, the instance uses the default query method.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.db.max.aggregate.size</td>
<td>Sets the maximum number of groups a grouped report or list renders. Larger values may affect system performance.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 100</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.db.max_view_records</td>
<td>Sets the maximum number of records returned when running a GlideRecord query in a script. Values larger than the default are not recommended as they may cause queries to consume excessive memory on the application server and can, in extreme cases, cause a system outage. This property does not control the maximum number of records that appear in a list, report, or exported file.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 10000</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.db.audit.ignore.delete                       | Specifies the tables where the sysAuditDelete file is not updated when records are deleted.  
- Type: string - a comma separated list of tables  
- Location: System Property [sys_properties] table |                                                                                                                                                                                                                                                                                                                                 |
| glide.discovery.application_mapping                | Enables or disables the Application Dependency Mapping (ADM) feature.  
- Type: string  
- Default value: true  
- Location: add to the System Property [sys_properties] table |                                                                                                                                                                                                                                                                                                                                 |
| glide.discovery.auto_adm                          | Automatically creates process classifiers for Application Dependency Mapping. When Discovery detects processes that are communicating over the network, “Pending Process” classifiers are automatically generated.  
- Type: string  
- Default value: true  
- Location: Add to the System Property [sys_properties] table |                                                                                                                                                                                                                                                                                                                                 |
| glide.discovery.log_message_chars                  | Specify the maximum length a log message can be before ServiceNow creates a preview for it in the list view. When a log message is longer than this value, ServiceNow creates a preview of the message with an ellipsis at the end of the message to indicate that there is content that is not shown. The preview size prevents any one list row from taking up the entire screen.  
- Type: integer  
- Default value: 200 (Characters)  
- Location: Discovery Definition > Properties |                                                                                                                                                                                                                                                                                                                                 |
| glide.discovery.use_cmdb_identifiers               | Controls whether Discovery uses the CMDB Identification and Reconciliation Framework, introduced with the Geneva release, or the legacy identifiers from previous releases.  
- Type: true/false  
- Default value: true  
- Location: Discovery Definition > Properties |                                                                                                                                                                                                                                                                                                                                 |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.domain.notify_change                      | Displays a notification message telling the user that the domain picker automatically changed  
• Type: true|false  
• Default value: true  
• Location: add to the System Property [sys_properties] table |
| glide.domain.notify_record_change               | Displays a notification message telling the user that the domain picker automatically changed because the record that the user is viewing changed the domain in which the user is in.  
• Type: true|false  
• Default value: false  
• Location: add to the System Property [sys_properties] table |
| glide.ecmdb.all_relationship_role              | An example value is: itil,asset,configuration.  
• Type: string  
• Default value: none  
• Location: add to the System Property [sys_properties] table |
| glide.email.append.timezone                    | Specifies whether to append the time zone to all dates and times in outbound emails.  
• Type: true | false  
• Default value: true  
• Location: System Properties > Email |
| glide.email.forward_subject_prefix             | Specifies the list of prefixes (comma-separated) in the subject line that identify a forwarded email.  
• Type: string  
• Default value: fw:,fwd:  
• Location: add to the System Property [sys_properties] table |
| glide.email.journal.lines                      | Specifies the number of entries from a journal field (such as Additional comments and Work notes) included in email notifications. A value of -1 includes all journal entries.  
• Type: integer  
• Default value: 3  
• Location: System Properties > Email |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.mail_to</td>
<td>Specifies the email address to send notifications that use the $mailto: variable.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: value of glide.email.user property</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.email.name_split</td>
<td>Delimiter between first and last name in an email address to identify users from incoming emails. For example, a delimiter of &quot;.&quot; (period) in the email address <a href="mailto:john.smith@company.com">john.smith@company.com</a> tells the system to look for a user record for John Smith.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: period (.)</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.email.notification.save_when_no_recipients</td>
<td>Controls whether a notification-generated sys_mail record is saved even if there are no recipients. Used in conjunction with other notification recipient logging properties, this property enables troubleshooting problems with notifications.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.email.override.url</td>
<td>Sets the URL to use in emailed links in place of the instance URL. The URL should end with nav_to.do. An example value is: <a href="https://servicenow.customerdomain.com/production/nav_to.do">https://servicenow.customerdomain.com/production/nav_to.do</a>.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: Instance URL</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.email.read.active</td>
<td>Specifies whether to enable or disable the inbound mail server.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.email.reply_subject_prefix | Specifies the list of prefixes (comma-separated) in the subject line that identify an email reply.  
  - Type: string  
  - Default value: re:aw:,r:  
  - Location: add to the System Property [sys_properties] table |
| glide.email.server | Specifies the URL of the outgoing SMTP mail server. Also used as incoming (POP) mail server if one is not specified. This server must be accessible from the ServiceNow domain. SMTP requires port 25 unless you specify another port with glide.smtp.port. Leave this value blank to disable email.  
  - Type: string  
  - Default value: relay  
  - Location: System Properties > Email |
| glide.email.smtp.active | Specifies whether to enable or disable the outgoing mail server.  
  - Type: true | false  
  - Default value: true  
  - Location: System Properties > Email |
| glide.email.smtp.max_recipients | Specifies the maximum number of recipients the instance can list in the To: line for a single email notification. Notifications that would exceed this limit instead create duplicate email notifications addressed to a subset of the recipient list. Each email notification has the same maximum number of recipients.  
  - Type: integer  
  - Default value: 100  
  - Location: add to the System Property [sys_properties] table |
| glide.email.smtp.max_send | Specifies how many emails to send through each new SMTP connection. The instance establishes a new SMTP connection if there are more emails to send than the specified value.  
  - Type: integer  
  - Default value: 100  
  - Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.test.user</td>
<td>Specifies the comma-separated list of email addresses to which the instance sends all email messages. Typically used in non-production instances for testing purposes.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.email.text_plain.strip_xhtml</td>
<td>Indicates whether both outbound and inbound emails that are shown in comments convert the XML to plain text (true) or preserve the XML (false).</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.email.user</td>
<td>Specifies the email address you want to use for SMTP authentication. The string before the @ (such as helpdesk) is used as the incoming (POP) account name if one is not specified.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: <code>&lt;yourinstance&gt;@service-now.com</code></td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.email.username</td>
<td>Specifies the display name for the email address you use for SMTP mail.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: IT Service Desk</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.email.user_password</td>
<td>Specifies the password for the outgoing SMTP mail server. Also used as the incoming mail server password if one is not specified.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: encrypted password</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.email.watermark.visible</td>
<td>Indicates whether the watermark in email notifications is visible (true) or is wrapped in a hidden div tag (false).</td>
</tr>
<tr>
<td>Note:</td>
<td>Email clients that use the plain text version of the email still show the watermark.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.email_client.show_sms_option</td>
<td>Specify whether a check box appears in the email client for sending the message to the user's SMS device. If no SMS device exists, the email client sends the message to the primary email device.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.excel.max_cells</td>
<td>Sets the maximum number of cells in an Excel export.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 500000</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.excel.use_user_date_format</td>
<td>Determines whether Excel exports use the date/time format specified in a user’s profile (true). If false, exports use the instance date/time format defined by the glide.sys.date_format property.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.export.csv.charset</td>
<td>Specify the character set used to export CSV files. See Supported Character Encodings for a list of supported character encoding options.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: windows-1252</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.export.csv.raw.value</td>
<td>When true, raw database values are exported instead of the display values when you export to CSV. When false, display values are exported instead.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Import Export</strong></td>
</tr>
<tr>
<td>glide.glidesoap.proxy_host</td>
<td>Specify the proxy server hostname or IP address for SOAP clients.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: value of glide.http.proxy_host</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.glidesoap.proxy_port</td>
<td>Specify the port number for the proxy server for SOAP clients.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: value of glide.http.proxy_port</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.help.default.page</td>
<td>Sets the overall help URL for the system if you are using context-sensitive help. This URL is used when there is not any context-sensitive help available for the form, list, or record.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: <a href="http://wiki.servicenow.com/">http://wiki.servicenow.com/</a></td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.history.max_entries</td>
<td>Sets the number of characters to display as a preview of journal input fields.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 250</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.home.page      | Determines which page to load when a user selects a homepage from the banner.  
- Type: string  
- Default value: home_splash.do? sysparm_direct=true  
- Location: System Property [sys_properties] table |
| glide.home.refresh_disabled | Determines whether homepage refresh is disabled (true) or enabled (false).  
- Type: true | false  
- Default value: false  
- Location: add to the System Property [sys_properties] table |
| glide.home.refresh_intervals | Comma-separated list of refresh intervals available on homepages.  
- Type: string  
- Default value: 300,900,1800,3600  
- Location: add to the System Property [sys_properties] table |
| glide.html.escape_script | Determines whether JavaScript tags are enabled (true) or disabled (false) in HTML fields.  
- Type: true | false  
- Default value: true  
- Location: System Property [sys_properties] table |
| glide.htmlsanitize_all_fields | Determines whether all HTML fields are sanitized to remove unwanted code.  
- Type: true | false  
- Default value: true  
- Location: System Property [sys_properties] table |
| glide.http.connection_timeout | Specify the maximum number of milliseconds an outbound HTTP request (such as web services) has to finish processing before the connection times out.  
- Type: integer  
- Default value: 100000 (100 seconds)  
- Location: System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.http.proxy_bypass_list              | Specify the semicolon-separated list of addresses that bypass the proxy server. Use an asterisk as a wildcard character to specify all or part of an address.  
  • Type: string  
  • Default value: none  
  • Location: System Property [sys_properties] table |
| glide.http.proxy_host                     | Specify the proxy server hostname or IP address  
  • Type: string  
  • Default value: none  
  • Location: System Property [sys_properties] table |
| glide.http.proxy_ntdomain                 | Specify the domain used to authenticate the proxy server with NTLM authentication.  
  • Type: string  
  • Default value: none  
  • Location: System Property [sys_properties] table |
| glide.http.proxy_nthost                   | Specify the hostname used to authenticate the proxy server with NTLM authentication.  
  • Type: string  
  • Default value: none  
  • Location: System Property [sys_properties] table |
| glide.http.proxy_ntpassword               | Specify the password used to authenticate the proxy server with NTLM authentication.  
  • Type: string  
  • Default value: none  
  • Location: System Property [sys_properties] table |
| glide.http.proxy_ntusername               | Specify the username used to authenticate the proxy server with NTLM authentication.  
  • Type: string  
  • Default value: none  
  • Location: System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.proxy_port</td>
<td>Specify the port number for the proxy server</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.http.proxy_password</td>
<td>Specify the password used to authenticate the proxy server.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.http.proxy_username</td>
<td>Specify the username used to authenticate the proxy server.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.http.timeout</td>
<td>(Web Service Consumer Plugin) Specifies the maximum number of milliseconds to wait before an outbound transaction times out.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 175000 (175 seconds)</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.i18n.force_index</td>
<td>Specifies that all translated fields are indexed regardless of the value of the table attribute text_index_translation.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td>glide.imap.secure</td>
<td>Specifies whether to enable SSL encryption for connections to the IMAP server.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.imap.secure_port                 | Specifies the communications port for IMAP secure connections.  
  - Type: string  
  - Default value: 995  
  - Location: add to the System Property [sys_properties] table                                                                                     |
| glide.imap.tls                         | Specifies whether to start the IMAP server in Transport Layer Security (TLS) mode.  
  - Type: true | false  
  - Default value: false  
  - Location: add to the System Property [sys_properties] table                                                                                      |
| glide.import_set.preserve.leading.spaces | Specifies whether the import process preserves leading spaces in Excel data cells. When false, the import process removes leading spaces from Excel data cells. When true, the import process preserves leading spaces.  
  **Note:** The import process always removes trailing spaces from Excel data cells.  
  - Type: true | false  
  - Default value: false  
  - Location: add to the System Property [sys_properties] table                                                                                      |
| glide.import_set_row.dynamically_add_fields | Specifies whether an import set can add new columns to the staging table (true) or not (false). Instances that contain large numbers of import sets can sometimes become unresponsive when an import adds a column because the instance must alter every row in the staging table. In some cases, the database alter table action causes an outage. Setting this property to false prevents an import set from adding columns to the staging table and produces a log message. As a workaround, administrators can manually add a column to the staging table by creating a new dictionary entry and then reimporting the import set.  
  - Type: true | false  
  - Default value: false  
  - Location: add to the System Property [sys_properties] table                                                                                      |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ir.query_method</td>
<td>Sets the query method for global text search. Only the <strong>Simple query</strong> method is supported.</td>
</tr>
<tr>
<td></td>
<td>• Type: choice list</td>
</tr>
<tr>
<td></td>
<td>• Default value: simple</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Global Text Search</strong></td>
</tr>
<tr>
<td>glide.integration.session_timeout</td>
<td>Length of time, in minutes, that an inactive integration session is maintained before the session times out.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.invalid_query.returns_no_rows</td>
<td>Controls how invalid GlideRecord queries are handled. When this property is true, invalid queries always return no rows. When this property is false (default), if a query is incorrect, such as by including an invalid field name, the invalid part of the query condition is ignored and results are based on the valid part of the query. To override this logic at the session level, execute gs.getSession().setStrictQuery(false). To restore strict query, execute gs.getSession().setStrictQuery(true). To use this property, change the line gr.addQuery('table', arguments.length == 1 ? record.getRecordClassName() : tableName); in the Workflow script include to gr.addQuery('table', (tableName) ? tableName : record.getRecordClassName() );</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.itil.assign.number.on.insert</td>
<td>Controls whether a task number is generated and assigned on load (Create New) or on submit of the task. This feature helps prevent unused task numbers.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.jdbcprobeloader.retry</td>
<td>Sets the number of times a JDBC probe attempts to connect to a JDBC data source.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 60</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.jdbcprobeloader.retry_millis</td>
<td>Sets the number of milliseconds a JDBC probe waits between retry attempts to a JDBC data source.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5000</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.knowman.search_character_limit</td>
<td>Minimum number of characters required for knowledge search. Search terms with fewer than this number of characters return no results.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 3</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.knowman.search.default_language</td>
<td>(Knowledge Management Internationalization Plugin v2) Default language for knowledge articles. If empty, defaults to logged in user language.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: empty</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.knowman.show_language_option</td>
<td>(Knowledge Management Internationalization Plugin v2) Determines whether the Language box appears (true) or is hidden (false) on articles with multiple translations.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.knowman.show_language_option.roles</td>
<td>(Knowledge Management Internationalization Plugin v2) Comma-separated list of roles that can see the Language box on articles with multiple translations.</td>
</tr>
</tbody>
</table>
|                                                | • Type: string  
|                                                | • Default value: none  
|                                                | • Location: add to the System Property [sys_properties] table                                                                                     |
| glide.knowman.show_links                       | Determines whether the Link line appears (true) or is hidden (false) in a knowledge article.                                                                                                               |
|                                                | • Type: true | false  
|                                                | • Default value: true  
|                                                | • Location: System Property [sys_properties] table                                                                                               |
| glide.ldap.allow_empty_group                   | Determines whether all members can be removed from an Active Directory security group.                                                                                                                     |
|                                                | • Type: true | false  
|                                                | • Default value: true  
|                                                | • Location: add to the System Property [sys_properties] table                                                                                     |
| glide.ldap.binary_attributes                   | Comma-separated list of LDAP attributes that should be converted from binary format to encoded64 strings. If you set this property, only the values listed are converted. The most common attributes are objectSID and objectGUID. These converted values are unique and can be used as the coalesce field on the LDAP import mapping. If this property is blank, ServiceNow tries to map these binary attributes without the conversion and they are not guaranteed to be unique since they are not properly converted to string values. |
|                                                | • Type: string  
|                                                | • Default value: objectsid,objectguid  
|                                                | • Location: System Property [sys_properties] table                                                                                               |
| glide.ldap.paging                               | Enables (true) or disables (false) LDAP paging query support. LDAP paging is a more efficient LDAP querying method for environments with more than 1000 users.                                                       |
|                                                | • Type: true | false  
|                                                | • Default value: true  
<p>|                                                | • Location: System Property [sys_properties] table                                                                                               |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.list.filter_max_length</td>
<td>Sets a maximum character limit for the condition builder query.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 0</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table.</td>
</tr>
<tr>
<td>glide.live_feed.company_feedExclude_groups</td>
<td>Controls whether messages posted to a public group appear (true) or are omitted (false) on the Company Feed.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.live_services</td>
<td>(Chat plugin) Enables (true) or disables (false) Live Services, such as chat support.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.login.no_blank_password</td>
<td>Prevents (true) or allows (false) logins from users with blank passwords. Often, importing lists of users creates a large number of users with blank passwords. By default, this property is set to true on production instances.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.login.home</td>
<td>Sets the default homepage users see after login. If blank, the last page visited is used. The format is &lt;page&gt;.do</td>
</tr>
<tr>
<td></td>
<td>• Type: sting</td>
</tr>
<tr>
<td></td>
<td>• Default value: home.do</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.max_journal_list_size</td>
<td>Sets the maximum size, in megabytes, of journal input fields.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 10</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.notification.recipient.include_logging</td>
<td>Master switch to enable/disable logging all reasons a recipient was included. If false, no include logging is performed.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.notification.recipient.exclude_logging</td>
<td>Master switch to enable/disable logging all reasons a recipient was excluded. If false, no exclude logging is performed.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.notification.recipient.exclude_logging.device_inactive</td>
<td>Logs recipients who are excluded because their chosen notification device record is marked as inactive.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>The glide.notification.recipient.exclude_logging property in the System Properties &gt; Email module must be enabled to modify this property.</td>
</tr>
<tr>
<td>glide.notification.recipient.exclude_logging.device_schedule</td>
<td>Logs recipients excluded because the chosen notification device record's schedule field excludes it.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>The glide.notification.recipient.exclude_logging property in the System Properties &gt; Email module must be enabled to modify this property.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.notification.recipient.exclude_logging.event_creator          | Logs recipients who are excluded because they initiated the notification event, such as updating an incident record, and the **Send to Event Creator** check box is cleared on the notification record.  
  - Type: true | false  
  - Default value: true   
  The **glide.notification.recipient.exclude_logging** property in the **System Properties > Email** module must be enabled to modify this property. |
| glide.notification.recipient.exclude_logging.invalid_email          | Logs recipients who are excluded because the email address for that user is invalid, for example the @ is missing, or empty.  
  - Type: true | false  
  - Default value: true   
  The **glide.notification.recipient.exclude_logging** property in the **System Properties > Email** module must be enabled to modify this property. |
| glide.notification.recipient.exclude_logging.user_calendar_integration_disabled | Logs recipients of calendar invitations who are excluded because the **Calendar Integration** field is set to None on the user record.  
  - Type: true | false  
  - Default value: true   
  The **glide.notification.recipient.exclude_logging** property in the **System Properties > Email** module must be enabled to modify this property. |
| glide.notification.recipient.exclude_logging.user_inactive          | Logs recipients who are excluded because the **Active** check box is cleared on the user record.  
  - Type: true | false  
  - Default value: true   
  The **glide.notification.recipient.exclude_logging** property in the **System Properties > Email** module must be enabled to modify this property. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.notification.recipient.exclude_logging.user_notification_disabled | Logs recipients who are excluded because the Notification field is set to Disabled on the user record.  
  - Type: true | false  
  - Default value: true  
  The glide.notification.recipient.exclude_logging property in the System Properties > Email module must be enabled to modify this property. |
| glide.notification.recipient.include_logging | Enables or disables logging all reasons a recipient was included. This property is a master switch. If it is set to true, the subsequent properties dealing with the inclusion of logging are enabled. If it is set to false, none of the subsequent properties relating to the inclusion of logging are enabled. |
| glide.notification.recipient.include_logging.delegate | Logs recipients who are included because they are delegates of another user.  
  - Type: true | false  
  - Default value: true  
  The glide.notification.recipient.include_logging property in the System Properties > Email module must be enabled to modify this property. |
| glide.notification.recipient.include_logging.event_parm | Logs recipients who are included because they are in the parm1 or parm2 fields of the event record.  
  - Type: true | false  
  - Default value: true  
  The glide.notification.recipient.include_logging property in the System Properties > Email module must be enabled to modify this property. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.notification.recipient.include_logging.recipient_fields | Logs recipients who are included via a notification target record, such as an incident record, specified in the **Users/Groups in Field** field for the notification record. The recipient_fields are fields in the target record that contain a recipient to add. For example, if the record that triggered the notification is an incident, and the **assigned_to** field for the incident is listed in recipient_fields, that user is included as a recipient.  
  - Type: true | false  
  - Default value: true  
  
The `glide.notification.recipient.include_logging` property in the **System Properties > Email** module must be enabled to modify this property. |
| glide.notification.recipient.include_logging.recipient_groups.group_email | Logs recipients who are included in a group email for any group provided in the notification record’s recipient_groups or the event **parm1** or **parm2** field.  
  - Type: true | false  
  - Default value: true  
  
The `glide.notification.recipient.include_logging` property in the **System Properties > Email** module must be enabled to modify this property. |
| glide.notification.recipient.include_logging.recipient_groups.manager | Logs recipients who are included because they manage any group provided in the notification record’s recipient_groups or the event **parm1** or **parm2** field.  
  - Type: true | false  
  - Default value: true  
  
The `glide.notification.recipient.include_logging` property in the **System Properties > Email** module must be enabled to modify this property. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.notification.recipient.include_logging.recipient_groups        | Logs recipients who are included via membership in any group provided in the notification record recipient_groups or the event parm1 or parm2 field.                                                                                                                           | • Type: true | false  
  • Default value: true  
  The glide.notification.recipient.include_logging property in the System Properties > Email module must be enabled to modify this property.                                                                                 |
| glide.notification.recipient.include_logging.recipient_users         | Logs recipients who are included via notification record’s Users field (recipient_users).                                                                                                                                                                                     | • Type: true | false  
  • Default value: true  
  The glide.notification.recipient.include_logging property in the System Properties > Email module must be enabled to modify this property.                                                                                 |
| glide.notification.recipient.include_logging.subscription             | Logs recipients because they are subscribed via User Notification Preferences.                                                                                                                                                                                                  | • Type: true | false  
  • Default value: true  
  The glide.notification.recipient.include_logging property in the System Properties > Email module must be enabled to modify this property.                                                                                 |
| glide.pdf_export_from_form_list.show_report_attrs                   | Enables or disable displaying the PDF page header for all PDFs generated from a list.                                                                                                                                                                                    | • Type: true | false  
  • Default value: true  
  • Location: add to the System Property [sys_properties] table                                                                                                                                                    |
| glide.pdf.max_rows                                                  | Sets the maximum number of rows in an exported PDF file.                                                                                                                                                                                                                 | • Type: integer  
  • Default value: 5000  
  • Location: add to the System Property [sys_properties] table                                                                                                                                                    |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.pdf.font.size</td>
<td>Sets the font size for exported PDF files.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 8</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.pg.any_rejection_rejects</td>
<td>Controls the default process guide rejection handling. If this property is set to true, the first rejection rejects the entity. If false, all users must reject the approval.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.phone_number_e164.allow_national_entry</td>
<td>Determines whether users can enter phone numbers in the local format or whether they must enter phone numbers in international format. When true, users can enter phone numbers in the local format listed in the territory selector. When false, users must enter phone numbers in the international format listed in the territory selector.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.phone_number_e164.display_national</td>
<td>Determines how a Phone Number (E164) field displays phone numbers. When set to true or form, a Phone Number (E164) field displays phone numbers in a local format on forms but as an international format on lists. When set to all, a Phone Number (E164) field always displays phone numbers in a local format. When set to user, a Phone Number (E164) field only displays phone numbers in a local format when the phone number matches the locale setting of the current user.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.phone_number_e164.display_territory_selector | Determines whether to display the territory selector. Hiding the territory selector restricts users to entering only local or national phone numbers.  
• Type: true | false  
• Default value: true  
• Location: add to the System Property [sys_properties] table |
| glide.phone_number_e164.display_territory_text | Determines when a Phone Number (E164) field displays a territory label. When set to all, a Phone Number (E164) always displays the territory label. When set to national, a Phone Number (E164) displays the territory label only if the phone number is in local format. When set to read-only, a Phone Number (E164) displays the territory label in read-only mode regardless of whether the number is in local or global format. When set to read-only-national, a Phone Number (E164) displays the territory label in read-only mode only if the number is in local format. When set to list, a Phone Number (E164) displays the territory label in a list. When set to list-national, a Phone Number (E164) displays territory label in a list if the number is in national format. When set to none, a Phone Number (E164) does not display the territory label.  
• Type: string  
• Default value: read-only  
• Location: add to the System Property [sys_properties] table |
| glide.phone_number_e164.display_users_idd | Determines whether to display the international direct dialing prefix between the territory selector and the input box on forms.  
• Type: true|false  
• Default value: false  
• Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.phone_number_e164.strict</td>
<td>Determines whether all phone number fields must match the display format of the field's select territory. When true, the phone number input box displays a red line underneath phone numbers that do not match the territory format listed in the territory selector. Users cannot save an invalid phone number. When false, the phone number input box displays a green line underneath phone numbers that do not match the territory format listed in the territory selector. Users can save an invalid phone number, and the territory selector offers the option to select an Other / Unknown territory format.</td>
</tr>
<tr>
<td>glide.pop3.ignore_headers</td>
<td>Specifies the comma-separated list of email headers that cause the instance to ignore an email message. Use the format name:value to specify email header types and values. You can use a wildcard (<em>) for the subtype. For example, &quot;Content-Type:multipart/</em>; report-type=delivery-status,&quot; ignores emails containing a type of multipart and a parameter of report-type=delivery-status. For syntax specifications, see <a href="http://www.w3.org/Protocols/rfc1341/4_Content-Type.html">http://www.w3.org/Protocols/rfc1341/4_Content-Type.html</a>.</td>
</tr>
<tr>
<td>glide.pop3.ignore_senders</td>
<td>Specifies the comma-separated list of senders that cause the instance to ignore an email message. Enter only the name before the at (@) sign.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.pop3.ignore_subjects</td>
<td>Specifies the comma-separated list of strings that cause the instance to ignore an email message if they are present at the start of a subject line. These values are case-sensitive.</td>
</tr>
<tr>
<td></td>
<td>Type: string</td>
</tr>
<tr>
<td></td>
<td>Default value: out of office autoreply, undeliverable:, delivery failure:, returned mail:, autoreply</td>
</tr>
<tr>
<td></td>
<td>Location: [System Properties &gt; Email]</td>
</tr>
<tr>
<td>glide.pop3.parse_end</td>
<td>[Legacy] Text indicating the end of the email body section where the instance should parse name:value pairs to change field values when processing inbound email actions. This property is no longer required to set field values from the email body.</td>
</tr>
<tr>
<td></td>
<td>Type: string</td>
</tr>
<tr>
<td></td>
<td>Default value: none</td>
</tr>
<tr>
<td></td>
<td>Location: [System Properties &gt; Email]</td>
</tr>
<tr>
<td>glide.pop3.parse_start</td>
<td>[Legacy] Text indicating the beginning of the email body section where the instance should parse name:value pairs to change field values when processing inbound email actions. This property is no longer required to set field values from the email body.</td>
</tr>
<tr>
<td></td>
<td>Type: string</td>
</tr>
<tr>
<td></td>
<td>Default value: none</td>
</tr>
<tr>
<td></td>
<td>Location: [System Properties &gt; Email]</td>
</tr>
<tr>
<td>glide.pop3.password</td>
<td>Password for the POP3 server. Contact your mail server administrator for this value.</td>
</tr>
<tr>
<td></td>
<td>Type: string</td>
</tr>
<tr>
<td></td>
<td>Default value: same encrypted password in glide.email.user_password</td>
</tr>
<tr>
<td></td>
<td>Location: [System Properties &gt; Email]</td>
</tr>
<tr>
<td>glide.pop3.process_locked_out</td>
<td>Enables (true) or disables (false) the ability for locked out users to trigger inbound actions.</td>
</tr>
<tr>
<td></td>
<td>Type: true</td>
</tr>
<tr>
<td></td>
<td>Default value: false</td>
</tr>
<tr>
<td></td>
<td>Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.pop3.reply_separators</td>
<td>Specifies the comma-separated list of separators that cause the instance to disregard everything below the text string in the message body. This list is case sensitive.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: <code>\n\n-----Original Message-----\n\n_______\n\nFrom:</code></td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.pop3.secure</td>
<td>Controls whether the instance connects to the incoming POP mail server using SSL encryption.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.pop3.server</td>
<td>Specifies the URL of the incoming POP mail server.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: value of <code>glide.email.server</code></td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.pop3.user</td>
<td>Incoming mail account name for the POP3 server.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: <code>&lt;yourinstance&gt;</code></td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.pop3readerjob.create_caller</td>
<td>Controls the behavior when an instance receives an email from an email address not associated with a user record. If this property is set to true, ServiceNow creates a new user record for the email address and places that new user in the <strong>Caller</strong> field of any created tickets. If the property is set to false, ServiceNow associates the new ticket to the <strong>Guest</strong> user record.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.processor.json.row_limit</td>
<td>Specify the maximum number of rows a JSON query returns</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 250</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.product.help_url</td>
<td>Controls whether the help icon, help.gifx, appears in (true) or is omitted from (false) the welcome banner.</td>
</tr>
</tbody>
</table>
|                         | • Type: true | false  
|                         | • Default value: true  
|                         | • Location: System Property [sys_properties] table                                               |
| glide.product.help_show  | Stores the favicon image displayed in bookmarks, tabs, and the browser address bar.            |
|                         | • Type: image  
|                         | • Default value: favicon.ico?v=4  
|                         | • Location: System Properties > System                                                         |
| glide.product.icon       | Specifies text to use in place of "ServiceNow" in the browser’s top title bar (and in browser tabs). This text is duplicated in the banner to the right of the logo unless you add display: none; to the end of the value field within the glide.product.name.style property.  |
|                         | • Type: string  
|                         | • Default value: ServiceNow  
|                         | • Location: System Property [sys_properties] table                                               |
| glide.product.name.style | Specifies the CSS properties used to display the glide.product.name text in the banner to the right of the logo. To not display it, add display: none; to the end of the value field in this property.  |
|                         | • Type: string  
|                         | • Default value: padding-bottom: 0px; padding-top: 0px;  
<p>|                         | • Location: System Property [sys_properties] table                                               |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.quota.manager.debug</td>
<td>Controls whether to display (true) or hide (false) additional information related to the Quota Manager, such as running transactions, canceled transactions, and what quotas are matched to transactions.</td>
</tr>
</tbody>
</table>
|                                           | • Type: true | false  
|                                           | • Default value: true  
|                                           | • Location: add to the System Property [sys_properties] table |
| glide.quota.manager.heartbeat             | Sets the number of seconds between the start of each Quota Manager heartbeat. This value determines how often the Quota Manager checks for transactions exceeding a quota and how often it writes status in the log file. |
|                                           | • Type: integer  
|                                           | • Default value: 1  
|                                           | • Location: add to the System Property [sys_properties] table |
| glide.quota.manager.minimum_transaction_time | Sets the minimum number of seconds a transaction must run before the Quota Manager matches it to a transaction quota. ServiceNow recommends setting this value to at least 1 second because smaller values decrease performance and because transactions shorter than 1 second are probably not worth canceling. For optimal performance, set this value to the value of your most restrictive quota. For example, if your most restrictive quota cancels transactions longer than 1 minute, set the minimum transaction time to 60 seconds. |
|                                           | • Type: integer  
|                                           | • Default value: 1  
|                                           | • Location: add to the System Property [sys_properties] table |
| glide.remote_glide_record.max_count      | Control the maximum number of records that the GlideRecord query method returns when using the SOAP web service. ServiceNow primarily uses this property to control the records returned when using a Perl API GlideRecord query. |
|                                           | • Type: integer  
|                                           | • Default value: 250  
<p>|                                           | • Location: add to the System Property [sys_properties] table |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.report.new_home.heavy</td>
<td>Sets the number of performance-intensive reports that are displayed on the Heavy tab of the report_admin's Reports list.</td>
</tr>
<tr>
<td></td>
<td>• Type: Integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 25</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.report.new_home.most_used</td>
<td>Sets the number of most used reports that are displayed on the Most used tab of the report_admin's Reports list.</td>
</tr>
<tr>
<td></td>
<td>• Type: Integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 25</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.report.new_home.unused</td>
<td>Sets the number of unused reports that are displayed on the Unused tab of the report_admin's Reports list.</td>
</tr>
<tr>
<td></td>
<td>• Type: Integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 180</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.report.use_charting_v2</td>
<td>Enables (true) or disables (false) the v2 charting engine for report generation.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.report_home.group_report.show_usr_grp</td>
<td>Enables (true) or disables (false) the Reporting preferences link in the user's profile.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<p>| Name                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| glide.rest.debug          | Logs all stages of REST processing, including processing times.                                                                                                                                                                                                                                                                                                                                                                                                                                           | Location: add to the System Property [sys_properties] table |
|                           | • Type: true | false                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Type: true | false                                      |
|                           |             | Default value: false                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Default value: false |
|                           |             | Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                                                                                                                                                                       | Location: add to the System Property [sys_properties] table |
| glide.rollback.version    | Controls whether rollback behavior is used (true) or not (false).                                                                                                                                                                                                                                                                                                                                                                                                                                 | Type: true | false                                      | Location: add to the System Property [sys_properties] table |
|                           |             | Default value: true                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Default value: true |
| glide.rss.max_rows        | Controls the maximum number of records returned by the RSS Feed Generator.                                                                                                                                                                                                                                                                                                                                                                                                                               | Type: integer                                                                                                                                               |
|                           |             | Default value: 1000                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Location: add to the System Property [sys_properties] table |
| glide.sc.reset_cascade    | If true, forces variable cascading when navigating between pages in an order guide.                                                                                                                                                                                                                                                                                                                                                                                                                 | Type: true | false                                      | Location: add to the System Property [sys_properties] table |
|                           |             | Default value: false                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Default value: false |
| glide.schedules.repeat_nth| Controls how a schedule entry with Repeats set to Monthly and Monthly Type set to Day of the Week is defined. Choices are Week or Day. The Week option is defined as choosing a day of the month in the nth week selects the nth day of the month. The Day option is defined as choosing the nth day of the month selects the nth day of the month. | Type: choice list                                         | Location: sys_.properties table                      |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.schedules.fifth</td>
<td>Controls how a schedule entry that selects the fifth occurrence of a day in month behaves in a month containing only four occurrences of the day. Choices are <strong>Last</strong>, <strong>Next</strong>, and <strong>Strict</strong>. The <strong>Last</strong> option selects the last (fourth) day of the month. The <strong>Next</strong> option selects the first day of the next month. The <strong>Strict</strong> option skips the day completely. This property is only valid when the <code>glide.schedules.repeat_nth</code> property is set to <strong>Day</strong>.</td>
</tr>
<tr>
<td></td>
<td>• Type: choice list</td>
</tr>
<tr>
<td></td>
<td>• Default value: last</td>
</tr>
<tr>
<td></td>
<td>• Location: <code>sys_properties</code> table</td>
</tr>
<tr>
<td>glide.script.ccsi.ispublic</td>
<td>Provides privacy control over client-callable script includes that are accessed by public pages. When this property is set to <strong>false</strong>, all client-callable script includes are private.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property <code>[sys_properties] table</code></td>
</tr>
<tr>
<td>glide.script.log_level</td>
<td>Controls the behavior of <code>gs.log</code>. Set this property to <strong>none</strong> to disable glide script logging, or <strong>print</strong> to save log data to the filesystem instead of the database. Use the default value <strong>all</strong> to save all glide script logs to the database.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: all</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property <code>[sys_properties] table</code></td>
</tr>
<tr>
<td>glide.security.auto.resubmit.ajax</td>
<td>Automatically resubmits timed-out Ajax requests.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property <code>[sys_properties] table</code></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.security.checkacl.before.setvalue</td>
<td>Controls how the high security setting glide.security.strict.updates processes ACLs. When true, the instance checks ACL rules for all columns before applying any updates to a row. When false, the instance applies ACL rules as each column is processed in alphabetical order. This behavior may result in one update preventing another update from occurring. For example, suppose you create a custom ACL rule to prevent updates after a record is closed. If you attempt to close an incident and also add information to a custom resolution code field in the same update, the close action is applied first because it is controlled by the [incident.state] column and the custom field by the [incident.u_resolution_code] column. After the instance applies the close action, the custom ACL rule prevents further write updates to the row.</td>
</tr>
</tbody>
</table>
|                                                            | • Type: true | false  
|                                                            | • Default value: true  
|                                                            | • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                                    |
| glide.security.csrf.handle.ajax.timeout                    | Handles errors for timed out Ajax requests.                                                                                                                                                                                                                                                                                                                      |
|                                                            | • Type: true | false  
|                                                            | • Default value: true  
|                                                            | • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                                    |
| glide.security.csrf.strict.validation.mode                 | Enforces strict validation on CSRF tokens so that users cannot resubmit a request if the CSRF token does not match.                                                                                                                                                                                                                                                                                                   |
|                                                            | • Type: true | false  
|                                                            | • Default value: false  
|                                                            | • Location: System Property [sys_properties] table                                                                                                                                                                                                                                                                                                           |
| glide.security.diag_txns_acl                               | Controls who can view the stats.do, threads.do, and replication.do pages. When set to true, only administrators or users from a known IP address are allowed to view the pages. When set to false, all users have access to the pages.                                                                                                                                                                                                 |
|                                                            | • Type: true | false  
|                                                            | • Default value: false  
<p>|                                                            | • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                                    |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.security.mime_type.aliasset         | Creates customized mime type alias sets. For example, "image/png=image/x-png".  
  • Type: string  
  • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                            |
| glide.security.granular.create            | Requires users to have write access on all individual fields on a table before they can create a record in that table.  
  • Type: true | false  
  • Default value: false  
  • Location: System Property [sys_properties] table                                                                                                                                                                                                                                                                                                         |
| glide.security.file.mime_type.validation  | Enables (true) or disables (false) mime type validation for file attachments.  
  • Type: true | false  
  • Default value: false  
  • Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                                                                                          |
| glide.security.strict_elevate_privilege   | Forces all elevated roles to be treated equally for users with the administrator role. When enabled, administrators need to explicitly, manually elevate themselves to any roles that are marked as elevated.  
  When disabled, administrators only need to manually elevate to the security_admin role. Other roles are automatically granted to administrators.  
  • Type: true | false  
  • Default value: true for new instances, false for upgraded instances  
  • Location: For new instances, this property is available on the System Property [sys_properties] table. For upgrades, administrators must add this property to the System Property [sys_properties] table                                                                                                                                 |

© 2017 ServiceNow. All rights reserved. 1379
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.set_x_frame_options</td>
<td>Enables (true) or disables (false) the X-Frame-Options response header to SAMEORIGIN for all UI pages. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a <code>&lt;frame&gt;</code> or <code>&lt;iframe&gt;</code>. Set this property to true to avoid clickjacking attacks, by ensuring that CMS content cannot be embedded into other sites.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.short_poll_delay</td>
<td>(Chat plugin) Sets the short polling delay, in milliseconds, for XMPP requests. Polling is the method by which the browser gets information from the server to send instant messages in chat.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1000 (1 second)</td>
</tr>
<tr>
<td></td>
<td>• Location: Social IT &gt; Chat Administration &gt; Properties</td>
</tr>
<tr>
<td>glide.shortened_journal_length</td>
<td>Sets the number of characters to display as a preview of journal input fields.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 512000</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.smtp.auth</td>
<td>Specifies whether to authenticate the outgoing SMTP mail server with the credentials provided in the user name (glide.email.user) and password (glide.email.user_password) properties.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Email</td>
</tr>
<tr>
<td>glide.smtp.dateformat</td>
<td>Specify the date format to use for outgoing email notifications</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: date format listed in email sender's user record [sys_user.date_format].</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.smtp.default_retry</td>
<td>Enables (true) or disables (false) resending email when an unknown SMTP error code is encountered. The instance only recognizes the SMTP error codes defined in the glide.smtp.defer_retry_ids property.</td>
</tr>
</tbody>
</table>
|                             | • Type: true | false  
|                             | • Default value: true  
|                             | • Location: **System Properties > Email**                                                                                                                 |
| glide.smtp.defer_retry_ids  | Specifies the comma-separated list of SMTP error codes that force the instance to resend email.                                                                                                         |
|                             | • Type: string  
|                             | • Default value: 421,450,451,452  
|                             | • Location: **System Properties > Email**                                                                                                             |
| glide.smtp.encryption       | Specifies how to encrypt communications with the SMTP server.                                                                                                                                               |
|                             | • Type: choice  
|                             | • Default value: none  
|                             | • Location: **System Properties > Email**                                                                                                             |
| glide.smtp.fail_message_ids | Specifies the comma-separated list of SMTP error codes that prevent the instance from resending email.                                                                                                     |
|                             | • Type: string  
|                             | • Default value: 500,501,502,503,504,550,551,552,553,554  
|                             | • Location: **System Properties > Email**                                                                                                             |
| glide.smtp.port             | Communications port on which the instance listens for SMTP traffic.                                                                                                                                          |
|                             | • Type: integer  
|                             | • Default value: 25  
|                             | • Location: add to the System Property [sys_properties] table                                                                                          |
| glide.smtp.secure           | [Legacy] Specifies whether to enable SSL encryption for SMTP connections. Use glide.smtp.encryption (See Outgoing Mail Server settings) instead to specify how to encrypt communications with the SMTP server.             |
|                             | • Type: true | false  
|                             | • Default value: false  
<p>|                             | • Location: add to the System Property [sys_properties] table                                                                                          |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.smtp.timeformat                      | Specify the time format to use for outgoing email notifications  
  • Type: string  
  • Default value: time format listed in email sender's user record [sys_user.time_format].  
  • Location: add to the System Property [sys_properties] table |
| glide.smtp.tls                             | [Legacy] Enables (true) or disables (false) Transport Layer Security (TLS) encryption for outgoing mail. Use glide.smtp.encryption (See Outgoing Mail Server settings) instead to specify how to encrypt communications with the SMTP server.  
  • Type: true | false  
  • Default value: false  
  • Location: add to the System Property [sys_properties] table |
| glide.soap.default_security_policy         | Specifies the name of SOAP security policy the instance uses when enforcing Web Services-Security (WSS) for inbound requests.  
  • Type: string  
  • Default value: default security policy  
  • Location: add to the System Property [sys_properties] table |
| glide.soap.import_set_insert_serialized.<tablename> | Controls the processing of web service inserts. When this property is set to true, the instance processes multiple simultaneous inserts one at a time (serially across nodes) to ensure an accurate transform. Serialized processing slows down the speed at which the instance processes inserts. When this property is set to false, multiple simultaneous inserts into an import set table result in simultaneous transforms that may produce duplicate target records due to the coalesce value being created at the same time.  
  **Tip:** Only set this value to false to optimize for performance when the related transform map does not have a coalesce value that may be present simultaneously.  
  • Type: true | false  
  • Default value: true  
  • Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.soap.request_processing_timeout            | Sets the maximum number of seconds a SOAP request has to finish processing before the connection times out. This property computes a default value from the value of the property glide.http.timeout divided by 1000. There might be network infrastructure (such as proxy servers) in place that implements a shorter timeout. In this case, a socket timeout may occur unless this property is set to a shorter value. In general, you should set this property to a value several seconds less than the shortest socket inactivity timeout in effect anywhere in the network path between the client application and the ServiceNow instance.  
- Type: integer  
- Default value: 175 (value of glide.http.timeout divided by 1000)  
- Location: add to the System Property [sys_properties] table |
| glide.spell.dictionary.en                        | Sets the spell checker dictionary used in the system for English users. There are dictionaries available for Brazilian Portuguese, Dutch, English US, English UK, French, German, Italian, Portuguese, Russian, Spanish, and Thai.  
- Type: choice list  
- Default value: en.dic (English US)  
- Location: System Property [sys_properties] table |
| glide.spell.dictionary.max_matches               | Sets the maximum number of spelling errors the spell-check should detect. By default it finds only 10 spelling errors. The instance ignores any additional errors after reaching the maximum value.  
- Type: integer  
- Default value: 10  
- Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.sys.activity_using_audit_direct                      | [Not Supported] Controls whether the record's history is generated using the Audit table (true) or not (false).  
[Required] Set the value to **false** to generate history with History Sets.  
• Type: true | false  
• Default value: false  
• Location: System Property [sys_properties] table |
| glide.sys.audit_inserts                                    | Controls whether the Audit table audits inserts (true) or not (false).  
• Type: true | false  
• Default value: false  
• Location: System Property [sys_properties] table |
| glide.sys_reference_row_check                              | Controls whether the script conditions of Access Control Rules apply to a table's reference fields.  
• Type: true | false  
• Default value: false  
• Location: add to the System Property [sys_properties] table |
| glide.template.max_context                                 | Specifies the maximum number of templates displayed in a form's context menu. If more than this number are available, users can click **Apply Template** on the context menu to open the reference list of templates.  
• Type: integer  
• Default value: 15  
• Location: System Property [sys_properties] table |
| glide.translate.learn                                      | Enables (true) or disables (false) adding the current language suffix to UI elements such as labels and messages. Enabling the language suffix assists with translating new customizations.  
• Type: true | false  
• Default value: false  
• Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.ts.index.attachment.debug | When the value is set to **true**, enables log messages for exceptions that occur when indexing attachments (default is **false**). You can leave this property enabled during normal operations to capture stack trace information about any exceptions.  
  * Type: true | false  
  * Default value: false  
  * Location: add to the System Property [sys_properties] table |
| glide.ts.index.attachment.list_terms.debug | When the value is set to **true**, the system logs all indexed terms when an attachment is indexed (default is **false**).  
  [Recommended] For optimal performance, set this property to **false** during normal operations. Only enable this property when you are actively debugging an issue.  
  * Type: true | false  
  * Default value: false  
  * Location: add to the System Property [sys_properties] table |
| glide.ui.activity.displayname | Determines whether the activity formatter shows **name** values (true) or **user_name** values (false).  
  * Type: true | false  
  * Default value: false  
  * Location: add to the System Property [sys_properties] table |
| glide.ui.activity.email_roles | Specifies the list of roles (comma-separated) that can view email in the Activity Formatter.  
  * Type: string  
  * Default value: itil  
  * Location: add to the System Property [sys_properties] table |
| glide.ui.activity.email.use_display | Specifies whether to display email addresses or the user IDs (display value of the User table) in email headers. If true, the instance searches for a user record with a matching email address. If it cannot find a matching user record, it displays the email address.  
  * Type: true | false  
  * Default value: false  
  * Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.activity_stream.form_button</td>
<td>Removes the activity stream button from forms.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.activity_stream.list_button</td>
<td>Removes the activity stream button from lists.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.activity_stream.style.comments</td>
<td>Changes the color of the left bar in activity stream comments in UI16.</td>
</tr>
<tr>
<td></td>
<td>• Type: color entry, either a name, such as blue or a code, such as #0000FF.</td>
</tr>
<tr>
<td></td>
<td>• Default value: transparent</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.activity_stream.style.work_notes</td>
<td>Changes the color of the left bar in activity stream work notes in UI16.</td>
</tr>
<tr>
<td></td>
<td>• Type: color entry, either a name, such as blue or a code, such as #0000FF.</td>
</tr>
<tr>
<td></td>
<td>• Default value: gold</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.activity.style.comments</td>
<td>Changes the background color of the activity stream comments in UI15.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: background-color: background-color: WhiteSmoke</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.activity.style.work_notes</td>
<td>Changes the background color of the activity stream work notes in UI15.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: background-color: LightGoldenRodYellow</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.allow_deep_html_validation</td>
<td>Allows administrators to prevent users from saving invalid HTML in a journal field.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.attachment.force_download_all_mime_types</td>
<td>Forces download of all attachment files.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.audit_deleted_tables</td>
<td>Comma-separated list of system tables for which the audit history tracks deletions.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: sys_user, sys_user_group, sys_user_role, sys_user_has_role, sys_user_gmember, sys_group_has_role, sys_security_acl_role</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.auto.recovery</td>
<td>Allows users to recover unsaved changes while working in the developer studio.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>Auto Recovery &gt; Properties</strong></td>
</tr>
<tr>
<td>glide.ui.auto.recovery.exclude.field.types</td>
<td>Comma-separated list of field types you want to exclude from automatic recovery.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>Auto Recovery &gt; Properties</strong></td>
</tr>
<tr>
<td>glide.ui.auto.recovery.unsupported.field.types</td>
<td>Comma-separated list of field types excluded from automatic recovery.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: password,password2,glide_encrypted,video,user_image,image</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.ui.auto.recovery.unsupported.tables | Comma-separated list of tables you want to exclude from automatic recovery.  
  • Type: string  
  • Default value: v_ws_editor  
  • Location: Auto Recovery > Properties |
| glide.ui.auto_req.extend.session | When set to true, this property enables the user to extend their user session by selecting a homepage refresh time. When set to false, it enforces session timeout. The session timeout value is ignored when the user specifies an automatic refresh value. For example, if the user selects 5 minutes for automatic homepage refresh, the session is renewed every five minutes. By adding this property and setting the value to false, administrators can force the user session to time out even if the user's page refreshes every x number of minutes. The user's session times out after the value specified in the session timeout, plus the selected refresh value. This property takes effect when the Remember me check box is not selected for the user. Tablet and mobile devices do not support this feature.  
  • Type: true|false  
  • Default value: true  
  • Location: add to the System Property [sys_properties] table |
| glide.ui.buttons_bottom | Controls whether UI actions appear at both the bottom and top of the form (true) or only at the top (false). This property works in UI11 only.  
  • Type: true | false  
  • Default value: false  
  • Location: System Property [sys_properties] table |
| glide.ui.cert_task_activity.fields | Defines which journal field is the task activity field.  
  • Type: string  
  • Default value: work_notes  
  • Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.chart.bar.horiz.max_col_slant_labels</td>
<td>Sets the maximum number of columns in a horizontal bar chart before slanting (angling) the labels.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.chart.height</td>
<td>Specifies the height of a chart in pixels.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 300</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.chart.pie.labels</td>
<td>Enables (true) or disables (false) labels on pie chart slices.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.chart.pie.labels.max_items</td>
<td>Sets the maximum number of pie chart slices on which to display labels.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 8</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.chart.width</td>
<td>Specifies the width of a chart in pixels.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 500</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.clickthrough.popup</td>
<td>Enables (true) or disables (false) displaying the pop-up diamond icon for a reference field and opening a new window when the icon is clicked. Note that related lists do not appear on forms opened in the pop-up window.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.clickthrough.replace</td>
<td>Enables (true) or disables (false) both the pop-up and clickthrough icons for reference fields.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.date_format.first_day_of_week</td>
<td>Specifies which day of the week that weeks start on for calendar reports. Values: 1=Sunday, 2=Monday.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.date_picker.first_day_of_week</td>
<td>Specifies the first (leftmost) day of the week for the date and date/time picker (1=Sunday, 2=Monday...).</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.default.applications</td>
<td>Comma-separated list of application names that open by default in the navigation pane when nothing is opened via user preferences. If the property is specified and is blank, no applications are opened in the navigation pane when no applications are opened via user preferences. If the property is not specified, the first application that is authorized for that user opens if no applications are opened via user preferences.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: first authorized application for the user</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.dirty_form_support</td>
<td>Enables (true) or disables (false) display of a confirmation message when a form has unsaved changes and the user leaves the form through any means except a submit (such as using the green back arrow, any form button, or other). This property is not supported in Safari.</td>
</tr>
<tr>
<td></td>
<td>- Type: true</td>
</tr>
<tr>
<td></td>
<td>- Default value: true</td>
</tr>
<tr>
<td></td>
<td>- Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.doctype</td>
<td>Enables or disables the UI15 interface.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.ui.email_client.autocomplete.count</td>
<td>Sets the maximum number of auto-complete matches the Email Client displays.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.ui.email_client.autocomplete.group</td>
<td>Specifies whether groups are included in auto-complete results for the Email Client.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.ui.email_client.email_address.disambiguator</td>
<td>Sets the columns from the User [sys_user] table that the auto-complete list displays. Separate each column name with a semicolon character (;). See the system dictionary for a list of available column names. For example, add the sys_user.email and sys_user.company columns to show a user's email address and company in the auto-complete list.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.ui.email_client.from | Specifies whether to display the **From:** line in the Email Client. Users can change the email address in the **From:** line by entering a new value.  
  - Type: true | false  
  - Default value: false  
  - Location: [System Properties > UI Properties](#) |
| glide.ui.email_client.reply_to | Specifies whether to display the **Reply to:** line in the Email Client.  
  - Type: true | false  
  - Default value: false  
  - Location: [System Properties > UI Properties](#) |
| glide.ui.filter.first_day_of_week | Identifies the first day of the calendar week for the company. By default, the start of the week is Monday, meaning that the calendar week begins with Monday and ends with Sunday. To change this behavior, add the property `glide.ui.filter.first_day_of_week` to the instance as an integer property. Set the value to the integer corresponding with the day of the week that the calendar begins on, where 1 is Sunday, 2 is Monday, and so on. The function impacts all charts and calculations where the day of the week is used as a parameter.  
  - Type: integer  
  - Default value: 2  
  - Location: add to the System Property [sys_properties] table |
| glide.ui.first.field.reference | Enables (true) or disables (false) having the first column in a list always link to the underlying record, even if it is a reference field. For example, if the first column on an Incident list is **Assigned to**, that value links to the Incident if this property is set to true.  
  - Type: true | false  
  - Default value: false  
  - Location: add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.form_annotations</td>
<td>Enables (true) or disables (false) form annotations, which allow you to add Custom, Section Separator, and Line Separator to a form.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.form_multiple_splits</td>
<td>Enables (true) or disables (false) multiple splits and end splits in the slushbucket on Personalize Form.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.glide_list.start.locked</td>
<td>Controls whether a glide_list (like the watch list) starts out locked (true) or unlocked (false) on a form.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.goto_use_contains</td>
<td>Controls whether the Go to navigation performs a &quot;contains&quot; query (true) or a &quot;greater than&quot; query (false) by default.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.homepage.parallel</td>
<td>Enables (true) or disables (false) use of parallel rendering.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.homepage.parallelism</td>
<td>Sets the maximum number of threads that should cooperate on rendering any given homepage.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 2</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.homepage.preview</td>
<td>Enables (true) or disables (false) displaying the preview icon for lists on the homepage.</td>
</tr>
<tr>
<td></td>
<td>- Type: true</td>
</tr>
<tr>
<td></td>
<td>- Default value: true</td>
</tr>
<tr>
<td></td>
<td>- Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.html.editor</td>
<td>Determines which HTML field editor to use, TinyMCE or htmlArea (legacy).</td>
</tr>
<tr>
<td></td>
<td>- Type: string</td>
</tr>
<tr>
<td></td>
<td>- Default value: tinyMCE</td>
</tr>
<tr>
<td></td>
<td>- Location: System Properties &gt; UI Properties</td>
</tr>
<tr>
<td>glide.ui.html.editor.toolbar.line1</td>
<td>Configures the editing toolbar (first line) for HTML fields when the TinyMCE editor is enabled.</td>
</tr>
<tr>
<td></td>
<td>- Type: string</td>
</tr>
<tr>
<td></td>
<td>- Location: System Properties &gt; UI Properties</td>
</tr>
<tr>
<td>glide.ui.html.editor.toolbar.line2</td>
<td>Configures the editing toolbar (second line) for HTML fields when the TinyMCE editor is enabled.</td>
</tr>
<tr>
<td></td>
<td>- Type: string</td>
</tr>
<tr>
<td></td>
<td>- Location: System Properties &gt; UI Properties</td>
</tr>
<tr>
<td>glide.ui.html.image.allow_url</td>
<td>Enables (true) or disables (false) uploading an image via URL from HTML Fields.</td>
</tr>
<tr>
<td></td>
<td>- Type: true</td>
</tr>
<tr>
<td></td>
<td>- Default value: false</td>
</tr>
<tr>
<td></td>
<td>- Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.ui.html.toolbar | Configures the editing toolbar for HTML fields.  
• Type: string  
• Default value: list of buttons  
• Location: System Properties > UI Properties |
| glide.ui.incident_activity.fields | Defines what fields are visible in the activity formatter. If the activities are personalized, this property updates automatically.  
• Type: string  
• Default value: list of fields  
• Location: System Properties > UI Properties |
| glide.ui.incident_activity.max_addresses | Specifies the maximum number of addresses to list in an email audit record. If the number of addresses exceeds this limit, the instance truncates the list after the maximum value and displays an ellipsis character (...).  
• Type: string  
• Default value: 5  
• Location: add to the System Property [sys_properties] table |
| glide.ui.incident_alert_activity.fields | Incident alert activity formatter fields. This is the list of fields tracked from the incident alert form in the activity formatter.  
• Type: string  
• Default value: opened_by, work_notes, comments, severity, estd_distruption_time, actual_disruption_time  
• Location: System Property [sys_properties] table |
| glide.ui.inline.images | Enables (true) or disables (false) use of data URLs for report images.  
• Type: true | false  
• Default value: false  
• Location: Add to the System Property [sys_properties] table |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.js_error_notify</td>
<td>Displays client script errors to users with the client_script_admin role. Also displays a generic error message to other users who encounter a client script error.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; All Properties</strong></td>
</tr>
<tr>
<td>glide.ui.label.enable</td>
<td>Enables (true) or disables (false) using labels, such as Most Active, Most Recent, or user created.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.max_calendar_duration</td>
<td>Maximum number of days that a single calendar report entry can display.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 90</td>
</tr>
<tr>
<td></td>
<td>• Range of possible values: 90 to 400</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.m_agents</td>
<td>Comma-separated list of browser agents considered as mobile browsers for the Smartphone Interface. These browsers are directed to the mobile pages instead of the full browser pages.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: iphone,android_phone,IEMobile,Windows Phone,iPod,Windows CE,BlackBerry,BB10</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.mobile_agents</td>
<td>Comma-separated list of browser agents considered as mobile browsers for the Legacy Smartphone Interface. These browsers are directed to the mobile pages instead of the full browser pages.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: iPod, Windows CE, BlackBerry, Android, Opera Mini, IEMobile, Windows Phone, iphone</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.nav.stripe.select.maxchars</td>
<td>Numerical character limit for dropdown menu choices within the nav stripe. The nav stripe displays at the top of the page when using UI11.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: add the property</td>
</tr>
<tr>
<td>glide.ui.permitted_tables</td>
<td>Comma-separated list of system tables that can be reported on. By default, system tables (which start with &quot;sys&quot;) cannot be reported on unless they are listed in this property.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: list of sys_ tables</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.reference.readonly.clickthrough</td>
<td>Enables (true) or disables (false) reference pop-ups on read-only reference fields. Reference pop-ups and click-throughs are hidden by default if a client script, UI policy, or ACL makes the field read-only. Being able to see or click through to the target record is unrelated to the reference field being writable.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td></td>
<td>If set to false, the administrator can override the system setting for a specific field by adding the readonly_clickthrough=true attribute to the dictionary entry.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.remember_view</td>
<td>Enables (true) or disables (false) use of user preferences for the last view.</td>
</tr>
<tr>
<td>• Type: true</td>
<td>false</td>
</tr>
<tr>
<td>• Default value: true</td>
<td></td>
</tr>
<tr>
<td>• Location: System Property [sys_properties] table</td>
<td></td>
</tr>
<tr>
<td>glide.ui.remember.me.default</td>
<td>Controls whether the <strong>Remember me</strong> check box is selected (true) or cleared (false) by default.</td>
</tr>
<tr>
<td>• Type: true</td>
<td>false</td>
</tr>
<tr>
<td>• Default value: true</td>
<td></td>
</tr>
<tr>
<td>• Location: System Property [sys_properties] table</td>
<td></td>
</tr>
<tr>
<td>glide.ui.report_expand_header</td>
<td>Comma-separated list of roles that can expand or collapse the report header, where a report is defined and built. The default is public, meaning everyone.</td>
</tr>
<tr>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td>• Default value: public</td>
<td></td>
</tr>
<tr>
<td>• Location: <strong>System Properties &gt; UI Properties</strong></td>
<td></td>
</tr>
<tr>
<td>glide.ui.section508</td>
<td>Enables (true) or disables (false) rendering of alternate text in place of images.</td>
</tr>
<tr>
<td>• Type: true</td>
<td>false</td>
</tr>
<tr>
<td>• Default value: false</td>
<td></td>
</tr>
<tr>
<td>• Location: <strong>System Properties &gt; UI Properties</strong></td>
<td></td>
</tr>
<tr>
<td>glide.ui.show_live_feed_activity</td>
<td>Enables (true) or disables (false) live feed for a record in the activity formatter.</td>
</tr>
<tr>
<td>• Type: true</td>
<td>false</td>
</tr>
<tr>
<td>• Default value: false</td>
<td></td>
</tr>
<tr>
<td>• Location: <strong>System Properties &gt; UI Properties</strong></td>
<td></td>
</tr>
<tr>
<td>glide.ui.session_timeout</td>
<td>Sets the session timeout, in minutes.</td>
</tr>
<tr>
<td>• Type: integer</td>
<td></td>
</tr>
<tr>
<td>• Default value: 30</td>
<td></td>
</tr>
<tr>
<td>• Location: add to the System Property [sys_properties] table</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.ui.stream_icon.<TABLENAME> | Toggles the displaying of the activity stream button on the specified table. For example: glide.ui.stream_icon.hr_case = true  
• Type: true | false  
• Default value: false |
| glide.ui.table.labels | Controls whether the system uses verbose labels for table names (true) or literal table names (false).  
• Type: true | false  
• Default value: true  
• Location: System Property [sys_properties] table |
| glide.ui.tablet_enabled | Enables (true) or disables (false) the tablet UI.  
• Type: true | false  
• Default value: true  
• Location: System Properties > Tablet UI Properties |
| glide.ui.tablet_agents | Uses the tablet UI if one of these strings (comma-separated) appears in the browser user_agent header  
• Type: string  
• Default value: ipad,android_tablet  
• Location: System Properties > Tablet UI Properties |
| glide.ui.tablet.title | Displays a brief page title for tablet UI.  
• Type: string  
• Default value: ServiceNow  
• Location: System Properties > Tablet UI Properties |
| glide.ui.tablet.title.ios_webapp | Displays the specified text as the default home screen icon label used in iOS version 6 and later.  
• Type: string  
• Default value: ServiceNow  
• Location: System Properties > Tablet UI Properties |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.task.insert</td>
<td>Enables (true) or disables (false) the use of <strong>Insert</strong> and <strong>Insert and Stay</strong> options on tables derived from Task (such as Incident, Change, and Problem).</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.textarea.character_counter</td>
<td>When true, displays a count of available characters for journal and multi-line text fields.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.textarea_initial_rows</td>
<td>Sets the number of rows initially displayed for multiline form elements. When the element is selected for editing, it is expanded. Set this value to 0 or blank to ignore this property. This functionality is used to conserve space on a form when multiline elements take up too much space on forms.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 0</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.update_on_iterate</td>
<td>Controls whether updates are saved (true) or discarded (false) when a user clicks the blue arrows on a form.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.ui_policy_debug</td>
<td>Enables (true) or disables (false) logging of UI policy processing in the Javascript debug window</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.welcome.profile_link</td>
<td>Enables (true) or disables (false) allowing users to click their name in the welcome message and see their user profile.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: In new instances true. In existing instances false.</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui11.show_switch_link</td>
<td>Displays (true) or hides (false) a banner link for switching between the UI11 and legacy interfaces.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.update.suppress_update_version</td>
<td>Comma-separated list of tables for which updates are not tracked in the Versions [sys_update_version] table. You cannot compare and revert versions for tables in this list.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: sys_user,sys_import_set_row</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.user.default_password</td>
<td>Default password for new users created from incoming email. Users must reset the password at first login.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: password</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.userauthgate.extauth.check</td>
<td>Enables the UserAuthenticationGate checks for external authentication mechanisms as well as internal ones.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.user.trusted_domain</td>
<td>Specifies the comma-separated list of trusted domains for creating users from incoming emails. Use an asterisk (*) to trust all domains. The instance ignores incoming email from other domains unless it is from an existing user’s address. The instance does not create guest users from email untrusted domains.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: *</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Email</strong></td>
</tr>
<tr>
<td>glide.workflow.model.cache.max</td>
<td>Maximum number of models held in the workflow cache. You must restart the instance after changing this property to apply the change.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 300</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>Workflow &gt; Properties</strong></td>
</tr>
<tr>
<td>glide.wsdl.definition.use_unique_namespace</td>
<td>Enables (true) or disables (false) use of a unique WSDL namespace value when publishing a ServiceNow table through web services. When this property is set to true, the WSDL target namespace is <a href="http://www.service-now.com/">www.service-now.com/</a>&lt;table name&gt;.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.xmlhttp.excessive</td>
<td>Sets the number of items visible in the Available half of a many-to-many or one-to-many collection box (slushbucket).</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 100</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.xmlprocessor.use_unload_format</td>
<td>Enables (true) or disables (false) unloading the XML using the display_value format. To unload using this format, add the parameter useUnloadFormat=true to the request URL.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: add to the System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.wsdl.show_nillable    | Enables (true) or disables (false) allowing a WSDL element to be valid if it has no element content despite a `{content type}` which would otherwise require content.  
  - Type: true | false  
  - Default value: false  
  - Location: add to the System Property [sys_properties] table                                                                                                                                                                                                                     |
| google.maps.auto_close      | If true, automatically closes the current `info window` before opening a new one. Clear the check box to allow multiple pop-up windows to remain open at once.  
  - Type: True/false  
  - Default value: true  
  - Location: **System Properties > Google Maps**                                                                                                                                                                                                                                        |
| google.maps.client          | Client ID for Google Maps API for Business. This is the client ID authorizing production use of Google Maps API for Business. By default, this ID is provided by ServiceNow. ServiceNow may require you to purchase a separate Client ID based on your usage.  
  - Type: string  
  - Default value: gme-servicenow  
  - Location: **System Properties > Google Maps**                                                                                                                                                                                                                                        |
| google.maps.key             | Map key from Google, tied to the URL of the server. This is the private key authorizing development use of Google Maps.  
  - Type: string  
  - Default value: empty  
  - Location: **System Properties > Google Maps**                                                                                                                                                                                                                                        |
| google.maps.latitude        | Starting latitude of the map. This value determines the starting latitude displayed in Google Maps.  
  - Type: string  
  - Default value: 36.008522  
  - Location: **System Properties > Google Maps**                                                                                                                                                                                                                                        |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.maps.longitude</td>
<td>Starting longitude of the map. This value determines the starting longitude displayed in Google Maps.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: -95.221764</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.max_items</td>
<td>Maximum number of items to display on the map. This setting determines how many icons can be displayed on a map.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 500</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.table</td>
<td>Table used by the map. This setting names the table containing the name, longitude, and latitude fields.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: cmn_location</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.zoom</td>
<td>Starting zoom level of the map (1 is the lowest).</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: 4</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>help.base.default</td>
<td>(Context-Sensitive Help plugin) Sets the base URL for help contexts in which ServiceNow Wiki is false and an absolute URL is not specified.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| help.base.servicenow                      | (Context-Sensitive Help plugin) Sets the base URL for help contexts in which ServiceNow Wiki is true  
• Type: string  
• Location: System Property [sys_properties] table |
| mid.server.rba_debug_powershell           | Enables probe-level debugging for PowerShell probes. With this property enabled, PowerShell probes write detailed activity information to the MID Server log.  
• Type: true|false  
• Default value: false  
• Location: Orchestration > MID Server Properties |
| password_reset.activity_monitor.incident_threshold | Determines the maximum number of lockouts before an alert is issued.  
• Type: integer  
• Default value: 10 (lockouts)  
• Location: Password Reset > Properties |
| password_reset.activity_monitor.incident_window | Determines the amount of time that is used for recording and counting the number of user lockouts.  
• Type: integer  
• Default value: 60 (minutes)  
• Location: Password Reset > Properties |
| password_reset.captcha.ignore             | Enables and disables captcha functionality.  
• Type: true|false  
• Default value: false  
• Location: Password Reset > Properties  
The password reset application uses Google reCAPTCHA as the default CAPTCHA service. To use the base-system CAPTCHA, change the password_reset.captcha.google.enabled system property to false. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>password_reset.qa.num_enroll</td>
<td>Specifies the number of questions a user has to select and answer to be enrolled in the password reset program.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5 (questions)</td>
</tr>
<tr>
<td></td>
<td>• Location: Password Reset &gt; Properties</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This security question property can be overridden by adding a value for the num_enroll parameter in a security question verification.</td>
</tr>
<tr>
<td>password_reset.qa.num_reset</td>
<td>Specifies the number of questions a user has to answer to verify their identity during the password reset process.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 3 (questions)</td>
</tr>
<tr>
<td></td>
<td>• Possible values: integers that are less than the number specified for the num_enroll property.</td>
</tr>
<tr>
<td></td>
<td>• Location: Password Reset &gt; Properties</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This security question property can be overridden by adding a value for the num_reset parameter in a security question verification.</td>
</tr>
<tr>
<td>password_reset.request.max_attempt</td>
<td>Determines the number of password reset attempts a user has before they are locked out for a period determined by the value in max_attempt_window.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 3 (attempts)</td>
</tr>
<tr>
<td></td>
<td>• Location: Password Reset &gt; Properties</td>
</tr>
<tr>
<td>password_reset.request.max_attempt_window</td>
<td>Determines how long a user is locked out or prevented from changing their password after trying the maximum number of times.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1440 (minutes)</td>
</tr>
<tr>
<td></td>
<td>• Location: Password Reset &gt; Properties</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| password_reset.request.retry_window           | Determines length of time before the count for password reset attempts refreshes.  
• Type: integer  
• Default value: 1440 (minutes)  
• Location: **Password Reset > Properties** |
| password_reset.request.success_window         | Determines how long a user is locked out after they have successfully reset their password.  
• Type: integer  
• Default value: 10 (minutes)  
• Location: **Password Reset > Properties** |
| password_reset.sms.default_complexity         | Specifies the number of characters required for a user to reset their password.  
**Note:** This SMS code property can be overridden by adding a value for the complexity parameter in an SMS code verification.  
• Type: integer  
• Default value: 4 (digits)  
• Location: **Password Reset > Properties** |
| password_reset.sms.expiry                     | Determines the amount of time, in minutes, until the SMS code sent to the user expires.  
**Note:** This SMS code property can be overridden by adding a value for the expiry parameter in an SMS code verification.  
• Type: integer  
• Default value: 5 (minutes)  
• Location: **Password Reset > Properties** |
| password_reset.sms.max_per_day                | Determines the maximum number of SMS codes that are sent to a user within one 24 hour period. When a user clicks the **Send Verification Code** button, the 24 hour period begins.  
• Type: integer  
• Default value: 10 (per day)  
• Location: **Password Reset > Properties**  
**Note:** This SMS code property can be overridden by adding a value for the max_per_day parameter in an SMS code verification.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>password_reset.sms.pause_window</strong></td>
<td>Determines the amount of time that needs to pass before another SMS code can be sent to a user.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This SMS code property can be overridden by adding a value for the <code>pause_window</code> parameter in an SMS code verification.</td>
</tr>
<tr>
<td></td>
<td>- Type: integer</td>
</tr>
<tr>
<td></td>
<td>- Default value: 2 (minutes)</td>
</tr>
<tr>
<td></td>
<td>- Location: <strong>Password Reset &gt; Properties</strong></td>
</tr>
<tr>
<td><strong>password_reset.wf.refresh_rate</strong></td>
<td>Determines how often to check status of the workflow. Represented in the password reset process progress bar.</td>
</tr>
<tr>
<td></td>
<td>- Type: integer</td>
</tr>
<tr>
<td></td>
<td>- Default value: 90000 (milliseconds)</td>
</tr>
<tr>
<td></td>
<td>- Location: <strong>Password Reset &gt; Properties</strong></td>
</tr>
<tr>
<td><strong>password_reset.wf.timeout</strong></td>
<td>Determines the maximum wait time, in milliseconds, for the workflow to execute. The workflow is triggered during the password reset request when the user clicks <strong>Submit</strong>.</td>
</tr>
<tr>
<td></td>
<td>- Type: integer</td>
</tr>
<tr>
<td></td>
<td>- Default value: 500 (milliseconds)</td>
</tr>
<tr>
<td></td>
<td>- Location: <strong>Password Reset &gt; Properties</strong></td>
</tr>
<tr>
<td><strong>sam.install_deletion_deadline</strong></td>
<td>Defines the number of days after which a software install is deleted if not discovered with the configuration item. The best practice is to use a value that is greater than the number of days between consecutive discovery runs.</td>
</tr>
<tr>
<td></td>
<td>- Type: integer</td>
</tr>
<tr>
<td></td>
<td>- Default value: 7</td>
</tr>
<tr>
<td></td>
<td>- Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td><strong>tablet.header.text</strong></td>
<td>Text that appears on the right-hand side of the tablet UI header.</td>
</tr>
<tr>
<td></td>
<td>- Type: string</td>
</tr>
<tr>
<td></td>
<td>- Default value: ServiceNow</td>
</tr>
<tr>
<td></td>
<td>- Location: <strong>System Properties &gt; Tablet UI Properties</strong></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tablet.footer.text</td>
<td>Text that appears on the footer of the tablet UI.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: Copyright ServiceNow 2012</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>System Properties &gt; Tablet UI Properties</strong></td>
</tr>
<tr>
<td>glide.bsm.map.style.text_color</td>
<td>Color of the text that appears under an unselected CI node.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: Black</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>BSM Map &gt; Map Properties</strong></td>
</tr>
<tr>
<td>glide.bsm.map.style.selection_text_color</td>
<td>Color of the text that appears under a selected CI node.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: White</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>BSM Map &gt; Map Properties</strong></td>
</tr>
<tr>
<td>glide.bsm.map.style.font_size</td>
<td>Font size of the text that appears with a CI node.</td>
</tr>
<tr>
<td></td>
<td>The default size is magnified for nodes with more connections and reduced for downstream nodes.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 14</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>BSM Map &gt; Map Properties</strong></td>
</tr>
<tr>
<td>glide.bsm.map.style.selection_background_color</td>
<td>Background color of a selected CI node. This color is also used with a node’s Highlight Hierarchy option.</td>
</tr>
<tr>
<td></td>
<td>• Type: color</td>
</tr>
<tr>
<td></td>
<td>• Default value: RoyalBlue</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>BSM Map &gt; Map Properties</strong></td>
</tr>
<tr>
<td>glide.bsm.max_levels</td>
<td>Maximum level depth from the root CI that can be initially displayed in Business Service Maps. Level depth is the graph distance between the root CI and a node. This value must be an integer.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer, valid values 1 to 10</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>BSM Map &gt; Map Properties</strong></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.bsm.map.style.font_family | Font family name used in the map text. If you designate a font that is not on your users’ system, the browser substitutes another font and the text may not render as you expect.  
  • Type: font name  
  • Default value: Arial  
  • Location: BSM Map > Map Properties |
| glide.bsm.new_node_color    | Color for nodes that became viewable from the last expand operation.  
  • Type: color  
  • Default value: PaleGreen  
  • Location: BSM Map > Map Properties |
| glide.bsm.too_many_children | Maximum number of child nodes to display. Nodes are collapsed for the map to meet this limit.  
  • Type: integer, valid values 1 or greater  
  • Default value: 10  
  • Location: BSM Map > Map Properties |
| glide.bsm.color.affect_neighbors | Color of an affected neighbor node. When a node has a service issue, all the nodes that are dependent on that node are considered affected nodes. In the map, the affected nodes are parents or grandparents of the node with the service issue.  
  • Type: color  
  • Default value: Beige  
  • Location: BSM Map > Map Properties |
| glide.bsm.max_nodes         | Maximum number of downstream nodes to retrieve from the database for a CI. If more nodes exist in the database, they are not displayed in the map.  
  • Type: integer  
  • Default value: 1000  
  • Location: BSM Map > Map Properties |
| glide.bsm.task_threshold    | Change the CI's glyph color from orange to red when the number of tasks reaches this threshold.  
  • Type: integer  
  • Default value: 3  
  • Location: BSM Map > Map Properties |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.bsm.refresh_interval</td>
<td>Seconds between each automatic reloading of troubles and tasks.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 30</td>
</tr>
<tr>
<td></td>
<td>• Range of possible values: 1 to 3600</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>BSM Map &gt; Map Properties</strong></td>
</tr>
<tr>
<td>glide.bsm.layout</td>
<td>Default layout for the BSM map. Options are Hierarchy, Radial, Grouping, and Force.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: Hierarchy</td>
</tr>
<tr>
<td></td>
<td>• Location: <strong>BSM Map &gt; Map Properties</strong></td>
</tr>
<tr>
<td>glide.entry.loggedin.page_ess</td>
<td>Specified the page that an ESS user, who by definition has no roles, is redirected to when the user logs in.</td>
</tr>
<tr>
<td></td>
<td>• Type: text</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
</tbody>
</table>

### Add a property

Administrators can add or create a property to control system behavior.

Some properties in the system are not visible in an instance by default and must be added to the System Property [sys_properties] table. If a feature requires the addition of a property, use one of the following methods:

- Add the property using [sys_properties.list]
- Create a properties module restricted to an administrator

**Important:** System properties store configuration information that rarely or never changes. Each time you change or add a system property, the system flushes the cache to keep all nodes in the cluster in synch. This cache flush has a very high performance cost for one to ten minutes, which can potentially cause an outage if done excessively. To prevent such outages, do not use a system property to store configuration information that changes more than once or twice a month. Instead, use a custom table to store regularly changing configuration information.

### Add a property using sys_properties.list

Use the [sys_properties.list] to add a system property.

1. Enter `sys_properties.list` in the Navigation filter.
   - The entire list of properties in the System Properties [sys_properties] table appears.
2. Verify the property does not already exist by searching for the property name.
3. Click **New**.
4. Complete the System Property form using the database name of the property. Make sure to specify the correct data **Type** and add the new value that you want the property to use.

Properties that you add already contain default values. You add properties to change this value.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type the name of the property you are creating.</td>
</tr>
<tr>
<td>Description</td>
<td>Type a brief, descriptive phrase describing the function of the property.</td>
</tr>
<tr>
<td>Choices</td>
<td>Comma-separated values for a choice list. If you need a different choice list label and value, use an equal sign (=) to separate the label from the value. For example, Blue=0000FF, Red=FF0000, Green=00FF00 displays Blue, Red, and Green in the list, and saves the corresponding hex value in the property value field.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the appropriate data type from the list (for example, integer, string, or true</td>
</tr>
<tr>
<td>Value</td>
<td>Set the desired value for property. All property values are stored as strings. When retrieving properties via the gs.getProperty() method, treat the results as strings. For example, a true</td>
</tr>
<tr>
<td>Ignore cache</td>
<td>Set this option to true to refresh the cache before processing the property. The cache stores commonly used items in memory such as forms and UI elements. Typically, you only ignore the cache if the system property depends upon a dynamic change on the form, and you want to ensure that the property uses the current value rather than a cached value.</td>
</tr>
<tr>
<td>Private</td>
<td>Set this property to true to exclude this property from being imported via update sets. Keeping system properties private prevents settings in one instance from overwriting values in another instance. For example, you may not want a system property in a development instance to use the same value as a production instance.</td>
</tr>
<tr>
<td>Read roles</td>
<td>Define the roles that have read access to this property.</td>
</tr>
<tr>
<td>Write roles</td>
<td>Defines the roles that have write access to this property.</td>
</tr>
</tbody>
</table>

The completed form looks similar to the following:
5. Submit the form.

Depending on the property name, an admin might be able to change its value only through this new Module. It may also appear in one of the Properties pages in System Properties. For example, any property whose name begins with `glide.ui` automatically appears in the **System Properties > UI Properties** page.

**Note:** If the **Ignore cache** check box is selected, the system flushes the server cache when the parameter is changed.

### Create a system properties module

You can add a module in the application navigator to access the list of system properties. This module makes it easy to add properties to the System Properties table.

1. Navigate to **System Definition > Application Menus**.
2. Search for the application you want to add the properties table to, for example **System Properties**. Select an application that is restricted to the admin role so that non-admin users cannot access it.
3. From the Modules related list, click **New**.
4. Complete the form fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Defines the module name. For example, <strong>All Properties</strong>.</td>
</tr>
<tr>
<td>Application Menu</td>
<td>Specifies the name of the application menu the module appears under. This field should automatically be populated with the name of the application you accessed the Modules related list from.</td>
</tr>
<tr>
<td>Link type</td>
<td>Specifies the type of link this module opens. For a list of system properties, select <strong>List of Records</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Table</td>
<td>Specifies the table used by the module. Select <strong>System Properties [sys_properties]</strong>.</td>
</tr>
</tbody>
</table>

5. Submit the form.
6. Verify that the module was created. For example, navigate to **System Properties > All Properties**.

**HTTP 500 error**

The HTTP 500 error is a generic error message that Internet Explorer passes to you when your web site errors for one reason or another; it provides no useful information.

We can change this setting, to allow you to see and read what the real error is. Good error information is essential to the debugging of any application, and the HTTP 500 error may in fact provide more information to the ServiceNow support team.

To enable Internet Explorer to show you the root cause of any error in your code, you need to make a change in the Internet Options window. You can access the Internet Options window by selecting the **Internet Options** item from the **Tools** menu at the top of Internet Explorer.

1. Select the Advanced tab.
2. Accessing the Advanced options displays a window that contains many settings that can be changed within Internet Explorer. These settings are broken down into categories. The category we are interested in is the Browsing Category.
3. Scrolling down the window will bring you to the **Show friendly HTTP error messages** checkbox. By default this box is checked in order to hide the real error from users. Once you have located this checkbox, remove the check and click the **OK** button.

You can now report any problem pages that you have had difficulty with and see the real error message that is being generated. Once you can see what the real error message is, then providing debugging information to the ServiceNow support team becomes so much easier.

**Query join and complexity size limits**

The ServiceNow ITSA Suite platform uses a relational database to store data. Retrieving data can involve multiple joins to create a single result set. While these joins are usually simple, in certain cases the system may issue very large joins to bring together large numbers (>20) of tables.
Database engines normally handle multiple joins quite well. The relational model assumes joins are cheap and efficient, and this is usually true. Rarely, however, exceedingly large joins may cause a database performance issue.

To mitigate this potential complexity, use the following property to limit join complexity and size. Navigate to System Properties > System to find the property with the description, Max number of database joins per query. Smaller values cause the system to issue a larger number of less complex queries. Larger values reduce the number of queries at the cost of additional complexity per query. In the absence of known database issues stemming from large join counts, this property should remain unchanged.

<table>
<thead>
<tr>
<th>Max number of database joins per query. Smaller values cause the system to issue a larger number of less complex queries. Larger values reduce the number of queries at the cost of additional complexity per query. In the absence of known database issues stemming from large join counts, this property should remain unchanged.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

**Figure 415: Query complexity limits**

This property specifies the maximum number of joins the system prefers to do. In certain edge cases, more complex queries might need to be issued, but generally no more than this number of joins will be issued. e.g., a value of 10 will result in no more than 10 tables being joined together in any given query.

The system must compensate for the fewer joins by issuing more queries to retrieve necessary data, so tuning this number down will result in more queries being sent to the database. In most cases, tuning this parameter is counterproductive.

**Note:** In the absence of known database issues stemming from large join counts, this parameter should remain unchanged.

### Tables and classes

If a table is not included in this list, it does not extend another table.

A table can "extend" another table.

A table that extends another table is called a child class, and the table it extends is the parent class.

If a table is extended, but itself is not extending another table, it is then called a base class.

For a child class record, the database stores new fields unique to the child class and stores the rest of the fields in the parent class. This means that one record in a child class exists both in the child class and the parent class. The record in the child class has a corresponding record (with the same sys_id identifier) in both the parent and child classes, which will always be accessed together as one record by the platform. When a record in one class is deleted, the platform automatically deletes its counterpart in its matching class.

**Important:** If you edit a field on a child table that is present on the parent table, you will also change it for the parent table and all other child tables.

To see the relationships between classes, use the schema map.

The following is a table of tables in the platform that extend other tables:

**Note:** If a table is not included here, it does not extend another table.
<table>
<thead>
<tr>
<th>Table Name and label</th>
<th>Description</th>
<th>Extension of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Software License [ast_license_adobe]</td>
<td>Adobe Software License</td>
<td>Base License Table [ast_license_base]</td>
</tr>
<tr>
<td>Generic Software License [ast_license_generic]</td>
<td>Generic Software License</td>
<td>Base License Table [ast_license_base]</td>
</tr>
<tr>
<td>Microsoft Software License [ast_license_msft]</td>
<td>Microsoft Software License</td>
<td>Base License Table [ast_license_base]</td>
</tr>
<tr>
<td>Symantec Software License [ast_license_symantec]</td>
<td>Symantec Software License</td>
<td>Base License Table [ast_license_base]</td>
</tr>
<tr>
<td>Service Contract [ast_service]</td>
<td>Information about service contracts for asset management purposes.</td>
<td>Contract [ast_contract]</td>
</tr>
<tr>
<td>Warranty [ast_warranty]</td>
<td>Warranty</td>
<td>Contract [ast_contract]</td>
</tr>
<tr>
<td>BSM Map Actions [bsm_action]</td>
<td>Actions on the map.</td>
<td>Diagrammer Actions [diagrammer_action]</td>
</tr>
<tr>
<td>Catalog Client Scripts [catalog_script_client]</td>
<td>Catalog Client Scripts</td>
<td>Client Script [sys_script_client]</td>
</tr>
<tr>
<td>Catalog UI Policy [catalog_ui_policy]</td>
<td>Catalog UI Policy</td>
<td>UI Policy [sys_ui_policy]</td>
</tr>
<tr>
<td>Change Phase [change_phase]</td>
<td>Change Phase</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Change Request [change_request]</td>
<td>Change Request</td>
<td>Task [task]</td>
</tr>
<tr>
<td>IMAC [change_request_imac]</td>
<td>IMAC</td>
<td>Change Request [change_request]</td>
</tr>
<tr>
<td>Change Task [change_task]</td>
<td>Change Task</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Accessory [cmdb_ci_acc]</td>
<td>Data about communication accessories such as phones and computers.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>AIX Server [cmdb_ci_aix_server]</td>
<td>AIX Server</td>
<td>Unix Server [cmdb_ci_unix_server]</td>
</tr>
<tr>
<td>Application [cmdb_ci_appl]</td>
<td>Data such as TCP port, PID and configuration files, related to running processes.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Application Software [cmdb_ci_application_software]</td>
<td>Data about installed software.</td>
<td>Software [cmdb_ci_spkg]</td>
</tr>
<tr>
<td>Application Server [cmdb_ci_app_server]</td>
<td>A base table for various application servers such as TomCat and Websphere.</td>
<td>Application [cmdb_ci_appl]</td>
</tr>
<tr>
<td>Domino [cmdb_ci_app_server_domino]</td>
<td>Domino</td>
<td>Application Server [cmdb_ci_app_server]</td>
</tr>
<tr>
<td>Jboss [cmdb_ci_app_server_jboss]</td>
<td>Jboss</td>
<td>Application [cmdb_ci_app_server]</td>
</tr>
<tr>
<td>Tomcat [cmdb_ci_app_server_tomcat]</td>
<td>Tomcat</td>
<td>Application Server [cmdb_ci_app_server]</td>
</tr>
<tr>
<td>BEA Weblogic [cmdb_ci_app_server_weblogic]</td>
<td>BEA Weblogic</td>
<td>Application Server [cmdb_ci_app_server]</td>
</tr>
<tr>
<td>IBM Websphere [cmdb_ci_app_server_websphere]</td>
<td>IBM Websphere</td>
<td>Application Server [cmdb_ci_app_server]</td>
</tr>
<tr>
<td>Circuit [cmdb_ci_circuit]</td>
<td>Circuit</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Cluster [cmdb_ci_cluster]</td>
<td>Cluster</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Cluster Node [cmdb_ci_cluster_node]</td>
<td>Cluster Node</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Cluster Virtual IP [cmdb_ci_cluster_vip]</td>
<td>Data about cluster VIP.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Communication Device [cmdb_ci_comm]</td>
<td>Data about communication devices such as cellphones, phones, conference phones and wifi.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Computer [cmdb_ci_computer]</td>
<td>Computer</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Computer Room [Roomcmdb_ci_computer_room]</td>
<td>Computer Room</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Computer Room AC [cmdb_ci_crac]</td>
<td>Computer room AC properties such as wattage and type of cooling.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Database [cmdb_ci_database]</td>
<td>Database</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Data Center [cmdb_ci_datacenter]</td>
<td>Data such as computers’ max power and power consumption, for the data center.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Database Catalog [cmdb_ci_db_catalog]</td>
<td>Database Catalog</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>DB2 Catalog [cmdb_ci_db_db2_catalog]</td>
<td>DB2 Catalog</td>
<td>Database Catalog [cmdb_ci_db_db2_catalog]</td>
</tr>
<tr>
<td>DB2 Instance [cmdb_ci_db_db2_instance]</td>
<td>DB2 Instance</td>
<td>Database Instance [cmdb_ci_db_db2_instance]</td>
</tr>
<tr>
<td>Database Instance [cmdb_ci_db_instance]</td>
<td>Data about database process.</td>
<td>Application [cmdb_ci_appl]</td>
</tr>
<tr>
<td>MSFT SQL Catalog [cmdb_ci_db_mssql_catalog]</td>
<td>Database catalog for MSFT SQL.</td>
<td>Database Catalog [cmdb_ci_db_mssql_catalog]</td>
</tr>
<tr>
<td>MSFT SQL Instance [cmdb_ci_db_mssql_instance]</td>
<td>Data about attributes specific for MSFT SQL instance.</td>
<td>Database Instance [cmdb_ci_db_mssql_instance]</td>
</tr>
<tr>
<td>MySQL Catalog [cmdb_ci_db_mysql_catalog]</td>
<td>MySQL Catalog</td>
<td>Database Catalog [cmdb_ci_db_mysql_catalog]</td>
</tr>
<tr>
<td>MySQL Instance [cmdb_ci_db_mysql_instance]</td>
<td>MySQL Instance</td>
<td>Database Instance [cmdb_ci_db_mysql_instance]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Oracle Catalog</td>
<td>Oracle Catalog</td>
<td>Database Catalog</td>
</tr>
<tr>
<td>[cmdb_ci_db_ora_catalog]</td>
<td></td>
<td>[cmdb_ci_db_catalog]</td>
</tr>
<tr>
<td>Oracle Instance</td>
<td>Data specific for Oracle.</td>
<td>Database Instance</td>
</tr>
<tr>
<td>[cmdb_ci_db_ora_instance]</td>
<td></td>
<td>[cmdb_ci_db_instance]</td>
</tr>
<tr>
<td>Sybase Catalog</td>
<td>Sybase Catalog</td>
<td>Database Catalog</td>
</tr>
<tr>
<td>[cmdb_ci_db_syb_catalog]</td>
<td></td>
<td>[cmdb_ci_db_catalog]</td>
</tr>
<tr>
<td>Sybase Instance</td>
<td>Data specific for Sybase.</td>
<td>Database Instance</td>
</tr>
<tr>
<td>[cmdb_ci_db_syb_instance]</td>
<td></td>
<td>[cmdb_ci_db_instance]</td>
</tr>
<tr>
<td>Desktop Software</td>
<td>Desktop Software</td>
<td>Software</td>
</tr>
<tr>
<td>[cmdb_ci_desktop_software]</td>
<td></td>
<td>[cmdb_ci_spkg]</td>
</tr>
<tr>
<td>Directory Server</td>
<td>Directory Server</td>
<td>Infrastructure Service</td>
</tr>
<tr>
<td>[cmdb_ci_directory_server]</td>
<td></td>
<td>[cmdb_ci_infra_service]</td>
</tr>
<tr>
<td>Disk</td>
<td>Data about physical disks on the server.</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_disk]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>Email Server</td>
<td>Email Server</td>
<td>Infrastructure Service</td>
</tr>
<tr>
<td>[cmdb_ci_email_server]</td>
<td></td>
<td>[cmdb_ci_infra_service]</td>
</tr>
<tr>
<td>ESX Server</td>
<td>Data about the physical ESX server where Vmware ESXi OS is running.</td>
<td>Server</td>
</tr>
<tr>
<td>[cmdb_ci_esx_server]</td>
<td></td>
<td>[cmdb_ci_server]</td>
</tr>
<tr>
<td>File System</td>
<td>Data such as mount point, capacity and type of file system for the file system information on the server.</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_file_system]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>NFS File System</td>
<td>Data about NFS file systems.</td>
<td>File System</td>
</tr>
<tr>
<td>[cmdb_ci_file_system_nfs]</td>
<td></td>
<td>[cmdb_ci_file_system]</td>
</tr>
<tr>
<td>SMB File System</td>
<td>SMB File System</td>
<td>File System</td>
</tr>
<tr>
<td>[cmdb_ci_file_system_smb]</td>
<td></td>
<td>[cmdb_ci_file_system]</td>
</tr>
<tr>
<td>FTP Server</td>
<td>FTP Server</td>
<td>Infrastructure Service</td>
</tr>
<tr>
<td>[cmdb_ci_ftp_server]</td>
<td></td>
<td>[cmdb_ci_infra_service]</td>
</tr>
<tr>
<td>Group</td>
<td>Group</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_group]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>HPUX Server</td>
<td>HPUX Server</td>
<td>Unix Server</td>
</tr>
<tr>
<td>[cmdb_ci_hpux_server]</td>
<td></td>
<td>[cmdb_ci_unix_server]</td>
</tr>
<tr>
<td>Infrastructure Service</td>
<td>Infrastructure Service</td>
<td>Application</td>
</tr>
<tr>
<td>[cmdb_ci_infra_service]</td>
<td></td>
<td>[cmdb_ci_appl]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>LDAP Service</td>
<td>LDAP Service</td>
<td>Infrastructure Service</td>
</tr>
<tr>
<td>[cmdb_ci_infra_service_ldap]</td>
<td></td>
<td>[cmdb_ci_infra_service]</td>
</tr>
<tr>
<td>Infrastructure Software</td>
<td>Infrastructure Software</td>
<td>Software</td>
</tr>
<tr>
<td>[cmdb_ci_inf Software]</td>
<td></td>
<td>[cmdb_ci_spkg]</td>
</tr>
<tr>
<td>IP Device</td>
<td>IP Device</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_ip_device]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>IP Network</td>
<td>Data such as subnet, router and router_interface_type, about IP networks.</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_ip_network]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>IP Phone</td>
<td>IP Phone</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_ip_phone]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>IP Router</td>
<td>A specialization of the Network Gear table.</td>
<td>Network Gear</td>
</tr>
<tr>
<td>[cmdb_ci_ip_router]</td>
<td></td>
<td>[cmdb_ci_netgear]</td>
</tr>
<tr>
<td>IP Server</td>
<td>IP Server</td>
<td>Infrastructure Service</td>
</tr>
<tr>
<td>[cmdb_ci_ip_server]</td>
<td></td>
<td>[cmdb_ci_infra_service]</td>
</tr>
<tr>
<td>IP Service</td>
<td>IP Service</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_ip_service]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>IP Switch</td>
<td>A specialization of the Network Gear table.</td>
<td>Network Gear</td>
</tr>
<tr>
<td>[cmdb_ci_ip_switch]</td>
<td></td>
<td>[cmdb_ci_netgear]</td>
</tr>
<tr>
<td>iPlanet Web Server</td>
<td>iPlanet Web Server</td>
<td>Web Server</td>
</tr>
<tr>
<td>[cmdb_ci_iplanet_web_server]</td>
<td></td>
<td>[cmdb_ci_web_server]</td>
</tr>
<tr>
<td>Linux Server</td>
<td>Linux Server</td>
<td>Server</td>
</tr>
<tr>
<td>[cmdb_ci_linux_server]</td>
<td></td>
<td>[cmdb_ci_server]</td>
</tr>
<tr>
<td>IBM Mainframe</td>
<td>IBM Mainframe</td>
<td>Server</td>
</tr>
<tr>
<td>[cmdb_ci_mainframe]</td>
<td></td>
<td>[cmdb_ci_server]</td>
</tr>
<tr>
<td>IBM Mainframe LPAR</td>
<td>IBM Mainframe LPAR</td>
<td>Server</td>
</tr>
<tr>
<td>[cmdb_ci_mainframe_lpar]</td>
<td></td>
<td>[cmdb_ci_server]</td>
</tr>
<tr>
<td>Microsoft IIS Web Server</td>
<td>Microsoft IIS Web Server</td>
<td>Web Server</td>
</tr>
<tr>
<td>[cmdb_ci_microsoft_iis_web_server]</td>
<td></td>
<td>[cmdb_ci_web_server]</td>
</tr>
<tr>
<td>Storage Device</td>
<td>Storage Device</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>[cmdb_ci_msd]</td>
<td></td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>Network Gear</td>
<td>Data about network equipment such as routers, switches, hubs, gateways and bridges.</td>
<td>Hardware</td>
</tr>
<tr>
<td>[cmdb_ci_netgear]</td>
<td></td>
<td>[cmdb_ci_hardware]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Netware Server</td>
<td>Netware Server</td>
<td>Server [cmdb_ci_server]</td>
</tr>
<tr>
<td>[cmdb_ci_netware_server]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Adapter</td>
<td>Data about network adapters such as netmasks and mac manufacturers.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_network_adapter]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Traffic</td>
<td>Network Traffic</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_net_traffic]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NGINX Web Server</td>
<td>NGINX Web Server</td>
<td>Web Server [cmdb_ci_web_server]</td>
</tr>
<tr>
<td>[cmdb_ci_nginx_web_server]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS/X Server</td>
<td>OS/X Server</td>
<td>Server [cmdb_ci_server]</td>
</tr>
<tr>
<td>[cmdb_ci_osx_server]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patch</td>
<td>Patch</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_patches]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDU</td>
<td>PDU</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_pdu]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outlet</td>
<td>Outlet</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_pdu_outlet]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Peripheral</td>
<td>Data such as monitors, docking stations, kvm switches, projectors, scanners, keyboards and UPS, related to computer peripherals.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_peripheral]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Printer</td>
<td>Data such as pages per minute (PPM), printer type, paper type and resolution, related to printers.</td>
<td>Hardware [cmdb_ci_hardware]</td>
</tr>
<tr>
<td>[cmdb_ci_printer]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print Queue</td>
<td>Print Queue</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_print_queue]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rack</td>
<td>Data such as rack units, rack units in use and power consumption, related to a rack in a datacenter.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_rack]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td>Additional data about servers, such as used for and server classification.</td>
<td>Computer [cmdb_ci_computer]</td>
</tr>
<tr>
<td>[cmdb_ci_server]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Service</td>
<td>Data such as used for, SLAs, business criticality and service classification, related to business service.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>[cmdb_ci_service]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>SNC Component</td>
<td>SNC Component</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Solaris Server</td>
<td>Solaris Server</td>
<td>[cmdb_ci_unix_server]</td>
</tr>
<tr>
<td>Software</td>
<td>Data such as version, install count, license count, package name, key (when SAM is enabled), related to software packages.</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Tomcat Connector</td>
<td>Tomcat Connector</td>
<td>[cmdb_ci]</td>
</tr>
<tr>
<td>Unix Daemon</td>
<td>Unix Daemon</td>
<td>IP Service [cmdb_ci_ip_service]</td>
</tr>
<tr>
<td>Unix Server</td>
<td>Unix Server</td>
<td>Server [cmdb_ci_server]</td>
</tr>
<tr>
<td>UPS</td>
<td>Data such as ups version, battery status and voltage parameters, related to UPS devices.</td>
<td>Hardware [cmdb_ci_hardware]</td>
</tr>
<tr>
<td>UPS Alarm</td>
<td>UPS Alarm</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>UPS Bypass</td>
<td>UPS Bypass</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>UPS Input</td>
<td>UPS Input</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>UPS Output</td>
<td>UPS Output</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Virtual Machine HyperVisor</td>
<td>Data such as VMWare/Parallels/Solaris, related to Virtual Machine hypervisors.</td>
<td>Application [cmdb_ci_appl]</td>
</tr>
<tr>
<td>Parallels</td>
<td>Parallels</td>
<td>Virtual Machine [cmdb_ci_vm]</td>
</tr>
<tr>
<td>VMware</td>
<td>VMWare specialization of Virtual Machine.</td>
<td>Virtual Machine [cmdb_ci_vm]</td>
</tr>
<tr>
<td>Zones</td>
<td>Zones</td>
<td>Virtual Machine [cmdb_ci_vm]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Virtual Private Network [cmdb_ci_vpn]</td>
<td>Virtual Private Network</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Websphere Cell [cmdb_ci_websphere_cell]</td>
<td>Websphere Cell</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>Web Application [cmdb_ci_web_application]</td>
<td>Data such as document base and application server, related to web applications.</td>
<td>Application [cmdb_ci_appl]</td>
</tr>
<tr>
<td>Web Server [cmdb_ci_web_server]</td>
<td>This is a specialization of the Server class for Windows.</td>
<td>Infrastructure Service [cmdb_ci_infra_service]</td>
</tr>
<tr>
<td>Web Site [cmdb_ci_web_site]</td>
<td>Data such as Web Server (Apache, IIS), related to Web sites.</td>
<td>Application [cmdb_ci_appl]</td>
</tr>
<tr>
<td>Zone [cmdb_ci_zone]</td>
<td>Zone</td>
<td>Configuration Item [cmdb_ci]</td>
</tr>
<tr>
<td>CI CPU Metrics [cmdb_metric_cpu]</td>
<td>CI CPU Metrics</td>
<td>CI Metric [cmdb_metric]</td>
</tr>
<tr>
<td>CI DB Connection Metric [cmdb_metric_db_connections]</td>
<td>CI DB Connection Metric</td>
<td>CI Metric [cmdb_metric]</td>
</tr>
<tr>
<td>Errors Logged [cmdb_metric_errors]</td>
<td>Errors Logged</td>
<td>CI Metric [cmdb_metric]</td>
</tr>
<tr>
<td>Events Processed [cmdb_metric_events_processed]</td>
<td>Events Processed</td>
<td>CI Metric [cmdb_metric]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Events Logged</td>
<td>Events Logged</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_event_logs]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI Java Metric</td>
<td>CI Java Metric</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_java]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>Linux Memory Metrics</td>
<td>Linux Memory Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_linux_memory]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI Load Metrics</td>
<td>CI Load Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_load]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI Log Count</td>
<td>CI Log Count</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_logs]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI MySQL Metrics</td>
<td>CI MySQL Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_mysql_statements]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI Oracle Metrics</td>
<td>CI Oracle Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_oracle]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>Partition Read/Write Statistics</td>
<td>Partition Read/Write Statistics</td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>[cmdb_metric_partition]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI Semaphore Metrics</td>
<td>CI Semaphore Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_semaphores]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI Service-Now Metrics</td>
<td>CI Service-Now Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_service_now]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>SQL Response Metrics</td>
<td>SQL Response Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_sql]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>CI Transaction Metrics</td>
<td>CI Transaction Metrics</td>
<td>CI Metric</td>
</tr>
<tr>
<td>[cmdb_metric_transactions]</td>
<td></td>
<td>[cmdb_metric]</td>
</tr>
<tr>
<td>Multiprobe</td>
<td>Multiprobe</td>
<td>Probe</td>
</tr>
<tr>
<td>[discovery_probes_multi]</td>
<td></td>
<td>[discovery_probes]</td>
</tr>
<tr>
<td>SNMP Probe</td>
<td>SNMP Probe</td>
<td>Probe</td>
</tr>
<tr>
<td>[discovery_probes_snmp]</td>
<td></td>
<td>[discovery_probes]</td>
</tr>
<tr>
<td>WMI Probe</td>
<td>WMI Probe</td>
<td>Probe</td>
</tr>
<tr>
<td>[discovery_probes_wmi]</td>
<td></td>
<td>[discovery_probes]</td>
</tr>
<tr>
<td>ECC Agent Counter Metric</td>
<td>ECC Agent Counter Metric</td>
<td>ECC Agent Metric</td>
</tr>
<tr>
<td>[ecc_agent_counter_metric]</td>
<td></td>
<td>[ecc_agent_metric]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>ECC Agent Memory Metric [ecc_agent_memory_metric]</td>
<td>ECC Agent Memory Metric</td>
<td>ECC Agent Metric [ecc_agent_metric]</td>
</tr>
<tr>
<td>ECC Agent RGR Metric [ecc_agent_rgr_metric]</td>
<td>ECC Agent RGR Metric</td>
<td>ECC Agent Metric [ecc_agent_metric]</td>
</tr>
<tr>
<td>ECC Agent Scalar Metric [ecc_agent_scalar_metric]</td>
<td>ECC Agent Scalar Metric</td>
<td>ECC Agent Metric [ecc_agent_metric]</td>
</tr>
<tr>
<td>Catalog Checkout Panel [expert_panel_catalog_checkout]</td>
<td>Catalog Checkout Panel</td>
<td>Wizard Panel [expert_panel]</td>
</tr>
<tr>
<td>Catalog Order [expert_panel_catalog_order]</td>
<td>Catalog Order</td>
<td>Wizard Panel [expert_panel]</td>
</tr>
<tr>
<td>KB Viewer [expert_panel_knowledge]</td>
<td>KB Viewer</td>
<td>Wizard Panel [expert_panel]</td>
</tr>
<tr>
<td>Record Generator [expert_panel_template]</td>
<td>Record Generator</td>
<td>Wizard Panel [expert_panel]</td>
</tr>
<tr>
<td>Wizard Client Script [expert_script_client]</td>
<td>Wizard Client Script</td>
<td>Client Script [sys_script_client]</td>
</tr>
<tr>
<td>Wizard Variable [expert_variable]</td>
<td>Wizard Variable</td>
<td>Question [question]</td>
</tr>
<tr>
<td>Import Log [import_log]</td>
<td>Import Log</td>
<td>Log Entry [syslog]</td>
</tr>
<tr>
<td>Computer [imp_computer]</td>
<td>Computer</td>
<td>Import Set Row [sys_import_set_row]</td>
</tr>
<tr>
<td>Location [imp_location]</td>
<td>Location</td>
<td>Import Set Row [sys_import_set_row]</td>
</tr>
<tr>
<td>Notification [imp_notification]</td>
<td>Notification</td>
<td>Import Set Row [sys_import_set_row]</td>
</tr>
<tr>
<td>User [imp_user]</td>
<td>User</td>
<td>Import Set Row [sys_import_set_row]</td>
</tr>
<tr>
<td>Incident [incident]</td>
<td>Incident</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Variable [item_option_new]</td>
<td>Variable</td>
<td>Question [question]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>KB Submission [kb_submission]</td>
<td>KB Submission</td>
<td>Task [task]</td>
</tr>
<tr>
<td>OLA [ola]</td>
<td>OLA</td>
<td>SLA [sla]</td>
</tr>
<tr>
<td>Plan MySQL [plan_mysql]</td>
<td>Plan MySQL</td>
<td>Plan Execution [plan_execution]</td>
</tr>
<tr>
<td>Plan Oracle [plan_oracle]</td>
<td>Plan Oracle</td>
<td>Plan Execution [plan_execution]</td>
</tr>
<tr>
<td>Problem [problem]</td>
<td>Problem</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Release Phase [release_phase]</td>
<td>Represents the stages of work required to complete a release.</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Release Task [release_task]</td>
<td>Represents the tasks required to create individual features.</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Scheduled Data Import [scheduled_data_import]</td>
<td>Scheduled Data Import</td>
<td>Scheduled Job [sysauto]</td>
</tr>
<tr>
<td>Scheduled Data Import [scheduled_import_set]</td>
<td>Scheduled Data Import</td>
<td>scheduled_data_import []</td>
</tr>
<tr>
<td>Dynamic Category [sc_category_top_n]</td>
<td>Dynamic Category</td>
<td>Category [sc_category]</td>
</tr>
<tr>
<td>Content Item [sc_cat_item_content]</td>
<td>Content Item</td>
<td>Catalog Item [sc_cat_item]</td>
</tr>
<tr>
<td>Execution Plan Approval Task [sc_cat_item_dt_approval]</td>
<td>Execution Plan Approval Task</td>
<td>Execution Plan Task [sc_cat_item_delivery_task]</td>
</tr>
<tr>
<td>Order Guide [sc_cat_item_guide]</td>
<td>Order Guide</td>
<td>Catalog Item [sc_cat_item]</td>
</tr>
<tr>
<td>Record Producer [sc_cat_item_producer]</td>
<td>Record Producer</td>
<td>Catalog Item [sc_cat_item]</td>
</tr>
<tr>
<td>Wizard Launcher [sc_cat_item_wizard]</td>
<td>Wizard Launcher</td>
<td>Catalog Item [sc_cat_item]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Request [sc_request]</td>
<td>Request</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Requested Item [sc_req_item]</td>
<td>Requested Item</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Catalog Task [sc_task]</td>
<td>Catalog Task</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Question [survey_question_new]</td>
<td>Question</td>
<td>Question [question]</td>
</tr>
<tr>
<td>Group Approval [sysapproval_group]</td>
<td>Group Approval</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Scheduled Email of Report [sysauto_report]</td>
<td>Scheduled Email of Report</td>
<td>Scheduled Job [sysauto]</td>
</tr>
<tr>
<td>Scheduled Script Extension [sysauto_script]</td>
<td>Scheduled Script Extension</td>
<td>Scheduled Job [sysauto]</td>
</tr>
<tr>
<td>Scheduled Entity Generation [sysauto_template]</td>
<td>Scheduled Entity Generation</td>
<td>Scheduled Job [sysauto]</td>
</tr>
<tr>
<td>Email Notification [sysevent_email_action]</td>
<td>Email Notification</td>
<td>Rule [sysrule]</td>
</tr>
<tr>
<td>Inbound Email Actions [sysevent_in_email_action]</td>
<td>Inbound Email Actions</td>
<td>Rule [sysrule]</td>
</tr>
<tr>
<td>Script Action [sysevent_script_action]</td>
<td>Script Action</td>
<td>Rule [sysrule]</td>
</tr>
<tr>
<td>Transaction Log Entry [syslog_transaction]</td>
<td>Transaction Log Entry</td>
<td>Rule [syslog]</td>
</tr>
<tr>
<td>Approval Rules [sysrule_approvals]</td>
<td>Approval Rules</td>
<td>Rule [sysrule]</td>
</tr>
<tr>
<td>Assignment Rules [sysrule_assignment]</td>
<td>Assignment Rules</td>
<td>Rule [sysrule]</td>
</tr>
<tr>
<td>Service Level Agreement [sysrule_escalate]</td>
<td>Service Level Agreement</td>
<td>Rule [sysrule]</td>
</tr>
<tr>
<td>Inactivity Monitor [sysrule_escalate_am]</td>
<td>Inactivity Monitor</td>
<td>Service Level Agreement [sysrule_escalate]</td>
</tr>
<tr>
<td>Table Name and label</td>
<td>Description</td>
<td>Extension of</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>View Rule [sysrule_view]</td>
<td>View Rule</td>
<td>Rule [sysrule]</td>
</tr>
<tr>
<td>Ticket [ticket]</td>
<td>Ticket</td>
<td>Task [task]</td>
</tr>
<tr>
<td>Variables [var_dictionary]</td>
<td>Variables</td>
<td>Dictionary [sys_dictionary]</td>
</tr>
<tr>
<td>Edit Field [v_field_editor]</td>
<td>Edit Field</td>
<td>Create Field [v_field_creator]</td>
</tr>
<tr>
<td>Edit Table [v_table_editor]</td>
<td>Edit Table</td>
<td>Create Table [v_table_creator]</td>
</tr>
<tr>
<td>Create Web Service [v_ws_creator]</td>
<td>Create Web Service</td>
<td>Create Table [v_table_creator]</td>
</tr>
<tr>
<td>Edit Web Service [v_ws_editor]</td>
<td>Edit Web Service</td>
<td>Create Web Service [v_ws_creator]</td>
</tr>
<tr>
<td>Create Web Service Field [v_ws_field_creator]</td>
<td>Create Web Service Field</td>
<td>Create Field [v_field_creator]</td>
</tr>
<tr>
<td>Edit Web Service Field [v_ws_field_editor]</td>
<td>Edit Web Service Field</td>
<td>Edit Field [v_field_editor]</td>
</tr>
<tr>
<td>Activity Variables [wf_activity_variable]</td>
<td>Variables for activities.</td>
<td>Variables [var_dictionary]</td>
</tr>
<tr>
<td>Workflow Log Entry [wf_log]</td>
<td>All of the events and history of the workflow.</td>
<td>Log Entry [syslog]</td>
</tr>
<tr>
<td>Workflow SC Variable [wf_variable]</td>
<td>The Service Catalog variables for a workflow.</td>
<td>Variable [item_option_new]</td>
</tr>
<tr>
<td>Workflow Schedule [wf_workflow_schedule]</td>
<td>Definitions of the times to run specific workflows.</td>
<td>Scheduled Script Execution [sysauto_script]</td>
</tr>
</tbody>
</table>

**Set up Section 508 compliance features**

Follow the instructions on this page to set up the ServiceNow Section 508 compliance features.

The ServiceNow platform includes features that support several specifications in the *Section 508 of the US compliance code* to make the interface accessible to users with disabilities. These features improve the user experience when using ServiceNow with screen readers and keyboard navigation.

- Enable the Section 508 accessibility feature.
- Disable the list editor feature.
• [Recommended] Customize the colors of mandatory field status indicators from the default settings to create greater contrast.

Enable the section 508 accessibility feature for UI16 and UI15

Alternative text is included for images throughout the user interface. Settings control the rendering of alternative text for screen readers in place of images and allows skip links to work more effectively.

Users can enable the Section 508 accessibility feature by selecting the Accessibility Enabled check box in the system menu. This enables accessibility for the current user.

System administrators can enable the Section 508 accessibility feature for all users:

1. Navigate to User Administration > User Preferences.
2. Select the glide.ui.accessibility preference.
3. Set the Value field to true.
4. Set the System field to true.

Enable the Section 508 accessibility feature for UI11

Alternative text is included for images throughout the user interface. Settings control the rendering of alternative text for screen readers in place of images and allows skip links to work more effectively.

The system administrator and users with the itil role can enable the glide.ui.section508 property and provide accessibility for all users.

1. Navigate to System Properties > UI Properties.
2. Select the check box for the following property: Controls rendering of alternate text. By default alternate text is NOT rendered (false).
3. Click Save.

Disable the list editor feature

The glide.ui.list_edit property controls list editing, which allows users to edit field values directly from a list without navigating to a form. When using the ServiceNow system with assistive technologies, disable this feature to allow users to tab between fields more quickly. The default value is true, which enables the list editor.

1. Navigate to System Properties > UI Properties.
2. Clear the check box for the Enable list editing property.
3. Click Save.

Change the color of mandatory field status indicators

Each element in a form can display a colored indicator to the left of the label. These field status indicators enable administrators to highlight certain fields and give status information to users.

The default indicators are:

• Red: mandatory field needs a value
• Light red: mandatory field has a value
• Green: field has changed
• Orange: your security access does not let you modify the field value

A system administrator can customize the colors of these field status indicators to be black and white or some other color combination with high contrast for easier viewing.
The field status indicator is accessible to assistive technologies such as screen readers, which read aloud the state of the field rather than the display color. The Alt and Tooltip fields are used to specify this alternative text.

See the following websites for information about CSS color declarations (including hex or RGB notation):

- HTML Colors (W3CSchools) for more information on hex and RGB notation.
- HTML Color Names (W3CSchools) for valid color names.

Use skip links

Skip links allow users to quickly navigate to the main content on a page, list, or form, bypassing icons, banner text, navigation links, and other elements.

The user interface includes skip links to the navigation menu, the content pane and, in UI11, to the form currently displayed in the content pane.

![Skip link image](image)

**Figure 416: Skip link**

- **Skip to Navigation Menu**: skips to the navigation filter in the application navigator.
- **Skip to Navigation Content**: skips to the first selectable field or link in the main content pane.
- **Skip to Form Content**: in UI11, skips to the first selectable field in the open form pane; if the form pane is not open, this link does not work.

**Attention**: You must enable the accessibility feature in order for skip links to work.

You may need to press the Tab key several times to move through the buttons in the Edge, the gray toolbar on the left side of the screen, to reach the skip links. When selected, skip links appear in the upper left corner of the user interface.

- Use the Tab and Shift+Tab keys to move forward and back through the skip links and other selectable fields.
- When a skip link is highlighted (outlined in blue), press Enter to select the link.

Tips for using search methods

There are several ways to search for information in the ServiceNow user interface.

- List views include a Go to search field in the list title bar. Using the * wildcard character in the search text allows you to perform a contains search rather than a starts with search. This type of query may take a little longer to run but can save keystrokes. You can also use several other wildcards with list searches.
• The navigation filter in the application navigator filters the applications and modules that appear in the navigator based on the entered text. To open a module, click the module name, or press the Down Arrow to highlight the module name, and then press Enter.

See *Available search options* on page 1068 for more information about these search methods.

**Web proxy**

Several ServiceNow properties support web proxy configuration.

**Basic proxy setup**

Use certain properties to manage connections to a ServiceNow instance through a proxy server.

Use the following properties to manage connections to a ServiceNow instance through a proxy server.

**Note:** Set these properties from the System Properties [sys_properties] table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.proxy_host</td>
<td>Specify the proxy server hostname or IP address.</td>
<td>proxy.company.com, 192.168.34.54</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
<td></td>
</tr>
<tr>
<td>glide.http.proxy_port</td>
<td>Specify the port number for the proxy server.</td>
<td>8080, 9100</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
<td></td>
</tr>
<tr>
<td>glide.http.proxy_username</td>
<td>Specify the username used to authenticate the proxy server.</td>
<td>proxyuser</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
<td></td>
</tr>
<tr>
<td>glide.http.proxy_password</td>
<td>Specify the password used to authenticate the proxy server.</td>
<td>password</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
<td></td>
</tr>
<tr>
<td>glide.email.override.url</td>
<td>Set the URL to use in emailed links in place of the instance URL. The URL should end with nav_to.do.</td>
<td><a href="https://servicenow.customerdomain.com/production/nav_to.do">https://servicenow.customerdomain.com/production/nav_to.do</a></td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value: instance URL</td>
<td></td>
</tr>
</tbody>
</table>
NTLM authentication

NTLM is the most complex of the authentication protocols supported by a basic web server.

NTLM is the most complex of the authentication protocols supported by a basic web server such as HttpClient. It is a proprietary protocol designed by Microsoft with no publicly available specification. Early versions of NTLM were less secure than Digest authentication due to faults in the design. However, these were fixed in a service pack for Windows NT 4 and the protocol is now considered more secure than Digest authentication.

NTLM authentication requires that an instance of NTCredentials be available for the domain name of the server or the default credentials. Since NTLM does not use the notion of realms, HttpClient uses the domain name of the server as the name of the realm. Also, the username provided to the NTCredentials should not be prefixed with the domain:

- Correct: adrian
- Incorrect: DOMAIN\adrian

There are some significant differences in the way NTLM works compared with basic and digest authentication. These differences are generally handled by HttpClient. However, having an understanding of these differences can help you avoid problems when using NTLM authentication.

- NTLM authentication works almost exactly the same as any other form of authentication in terms of the HttpClient API. The only difference is that you need to supply NTCredentials instead of UsernamePasswordCredentials (NTCredentials actually extends UsernamePasswordCredentials so you can use NTCredentials right throughout your application, if needed).
- The realm for NTLM authentication is the domain name of the computer being connected. This can be troublesome because servers often have multiple domain names. Only the domain name that HttpClient connects to, as specified by the HostConfiguration, is used to look up the credentials. While initially testing NTLM authentication, it is best to pass the realm in as null, which is used as the default.
- NTLM authenticates a connection and not a request. So you need to authenticate every time a new connection is made and keeping the connection open during authentication is vital. For this reason, NTLM cannot be used to authenticate with both a proxy server and the web server, nor can NTLM be used with HTTP 1.0 connections or web servers that do not support HTTP keep-alives.

Note: Set these properties from the System Properties [sys_properties] table.

Table 319: NTLM authentication

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.proxy_ntusername</td>
<td>Specify the username used to authenticate the proxy server with NTLM authentication.</td>
<td>username</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
<td></td>
</tr>
<tr>
<td>glide.http.proxy_ntpassword</td>
<td>Specify the password used to authenticate the proxy server with NTLM authentication.</td>
<td>password</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
<td></td>
</tr>
</tbody>
</table>
Proxy servers for SOAP clients

Administrators can specify separate proxy settings for SOAP clients, such as the MID Server or ODBC Driver.

To specify a proxy server for a MID Server, see MID Server configuration. To specify a proxy server for the ODBC driver, see ODBC via proxy servers on page 3488.

Bypass the proxy server

Administrators can configure ServiceNow to bypass the proxy server for specific URLs or URL patterns. Typically, internal addresses do not need a proxy server for SOAP communications.

Table 320: Bypass the proxy server

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.proxy_bypass_list</td>
<td>Specify the semicolon-separated list of addresses that bypass the proxy server. Use an asterisk as a wildcard character to specify all or part of an address.</td>
<td>127.0.0.1;*.internal.com;localhost</td>
</tr>
</tbody>
</table>

Data management

Data in ServiceNow is stored and managed according to a principled structure that administrators can view and configure. Data management functions include importing, exporting, and archiving database data and configuring fields and tables.

- The data is stored in database tables and records.
- Various tools exist to manage this data.
• The schema map displays CI relations graphically.
• Data dictionary tables store structure and relationship definitions.
• Various plugins provide additional functionality.
• Data import and export simplifies managing large sets of data through import maps and a number of exportable formats.
• Field normalization makes records more readable and prevents duplication of data.
• Scripting support allows for advanced customizations and automation of data management.

Database structure

All of the information in the instances is stored in tables, which consist of a series of records. The record in turn holds a series of fields that hold the individual bits of data and can be viewed either as a list or a form.

Tables can be related to each other in the following ways:

• **Extensions**: A table can extend another table. The table doing the extending (child class) includes all of the fields of the other table (parent class) and adds its own fields. For instance, the Incident [incident] table has all of the Task [task] table fields (because an incident is a special form of task) and has its own incident-specific tasks.

• **One-to-Many**: Within a table, a field can hold a reference to a record on another table. There are three types of one-to-many relationship fields:
  - Reference Field: allow a user to select a record on a table defined by the reference field. For instance, the **Caller** field on the Incident table allows the user to select any record on the User table.
  - Glide List: allows a user to select multiple records on a table defined by the glide list. For instance, the **Watch list** field on the Incident table allows the user to select records on the User table.
  - Document ID Field: allows a user to select a record on any table in the instance. These fields are much less common, but one example is the **Document** field on the Translated Text [sys_translated_text] table.

• **Many-to-Many**: Two tables can have a bi-directional relationship, so that the related records are visible from both tables in a related list.

• **Database Views**: Two tables can be joined virtually with Database Views to enable reporting on data that might be stored over more than one table.

Data management tools

There are a number of tools that can help manage data within the instance.

**Schema Map**

The Schema Map displays the relationships between tables visually, helping to navigate through the database structure.

The Schema Map provides an interface for viewing the relationships between tables. The inter-table relationships it captures include many-to-many relationships, tables that extend other tables, and tables that reference other tables through reference fields.

**Data Dictionary Tables**

Data dictionary tables holds information that defines the database and can be accessed for information on the database schema.
These tables hold important information on the database and its structure:

- **Tables [sys_db_object]**: contains a record for each table.
- **Dictionary Entries [sys_dictionary]**: contains additional details for each table and the definition for every column on each table. Each row represents either a column on a table or a table.
- **Field Labels [sys_documentation]**: contains the human-readable labels and language information.

**Table Cleaner**

Table cleaner automatically deletes records on certain tables to prevent data growing exponentially.

The system automatically deletes records from specific tables after a specific time to deletion. Deleting these records automatically prevents the tables from growing to an unmanageable size. The time before a record is deleted begins on the date and time value in the tracked field.

The Table Cleaner scheduled job runs the table cleaner every hour. To view the list of tables that are auto-cleaned, in the Navigator filter, type: `sys_auto_flush_list.do`.

All records with `MatchField < (current_time - Age in seconds)` are deleted.

- The **MatchField** field represents a Date/Time column in the table that you are trying to clean up.
- The **Age in seconds** field represents a value in seconds.

You can set up multiple table cleaner entries for a particular table. Performance depends on the size of the table and the conditions used. For example, if you use a custom column in a very large table that has no index on, performance is severely degraded. Performance also depends on the number of rows to be deleted.

**Table 321: Cleaned tables**

<table>
<thead>
<tr>
<th>Table</th>
<th>Tracked field</th>
<th>Time to deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys_poll</td>
<td>sys_created_on</td>
<td>1 day</td>
</tr>
<tr>
<td>cmdb_metric</td>
<td>sys_created_on</td>
<td>2 days</td>
</tr>
<tr>
<td>ecc_agent_metric</td>
<td>sys_created_on</td>
<td>30 days</td>
</tr>
<tr>
<td>v_transaction</td>
<td>sys_created_on</td>
<td>1 day</td>
</tr>
<tr>
<td>wf_context</td>
<td>ended</td>
<td>180 days</td>
</tr>
<tr>
<td>sys_cache_flush</td>
<td>sys_created_on</td>
<td>1 hour</td>
</tr>
<tr>
<td>sys_replication_queue</td>
<td>sys_created_on</td>
<td>1 day</td>
</tr>
<tr>
<td>sysevent</td>
<td>sys_created_on</td>
<td>7 days</td>
</tr>
<tr>
<td>ecc_queue</td>
<td>sys_created_on</td>
<td>30 days</td>
</tr>
<tr>
<td>syslog</td>
<td>sys_created_on</td>
<td>90 days</td>
</tr>
<tr>
<td>sys_querystat</td>
<td>sys_created_on</td>
<td>30 days</td>
</tr>
<tr>
<td>plan_execution</td>
<td>sys_created_on</td>
<td>30 days</td>
</tr>
<tr>
<td>sys_user_session</td>
<td>last_accessed</td>
<td>1 day</td>
</tr>
<tr>
<td>sys_user_preference</td>
<td>sys_updated_on</td>
<td>1 day</td>
</tr>
<tr>
<td>sys_history_set</td>
<td>sys_updated_on</td>
<td>30 days</td>
</tr>
<tr>
<td>Table</td>
<td>Tracked field</td>
<td>Time to deletion</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>------------------</td>
</tr>
<tr>
<td>label_history</td>
<td>sys_created_on</td>
<td>30 days</td>
</tr>
<tr>
<td>ts_search_summary</td>
<td>sys_created_on</td>
<td>30 days</td>
</tr>
<tr>
<td>sys_cluster_message</td>
<td>sys_created_on</td>
<td>1 day</td>
</tr>
<tr>
<td>cmdb_tcp_connection</td>
<td>sys_updated_on</td>
<td>30 days</td>
</tr>
<tr>
<td>cmdb_running_process</td>
<td>sys_updated_on</td>
<td>30 days</td>
</tr>
</tbody>
</table>

Data management plugins

There are a few plugins that extend the data management functionality.

Plugins that extend the data management functionality

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Archiving plugin</td>
<td>provides the ability to archive records to minimize performance issues</td>
</tr>
<tr>
<td>Database Rotations plugin</td>
<td>provides tools for managing large tables to minimize performance issues</td>
</tr>
<tr>
<td>Discovery plugin</td>
<td>provides the ability to auto-populate the CMDB with data about hardware, software, and networks</td>
</tr>
<tr>
<td>Domain Support plugin on page 1439</td>
<td>provides the ability to separate an instance into different distinct domains</td>
</tr>
<tr>
<td>Many to Many Task Relations plugin on page 1438</td>
<td>provides the ability to define many-to-many relationships between task tables</td>
</tr>
</tbody>
</table>

Data Rotations plugin

The Database Rotations plugin mitigates performance issues caused by large tables by breaking up overly-large tables into a series of smaller tables, through two methods:

- Table Rotation: sets up multiple tables for the same purpose that the platform uses on a rotating basis, deleting old information as it goes.
- Table Extension: creates new tables for the same purpose as the platform fills old tables.

Many to Many Task Relations plugin

The Many to Many Task Relations plugin allows administrators to manage the many-to-many relationships between task tables and record the relationship in more detail beyond Parent > Child.
Domain Support plugin

The Domain Support plugin provides a method for separating an instance into different domains, including partitioning data and securing the domains so that only members of the domain can see the data from their domain. The Data Separation part of the plugin can be implemented separately, if desired.

Data import and export tools

Tools are available for exporting to and importing data from an instance.

**Note:** In addition, data can be imported or exported with web services. LDAP data can be accessed with the LDAP integration in the base system.

Data export tools

The following tools are available to export data from an instance:

- Manually exporting from a record list.
- Exporting data on a schedule with scheduled reports.
- Exporting data with a URL parameter.
- Exporting data with a .NET application.

Data import tools

The following tools are available to import data from an instance:

- Manually importing from a record list.
- Importing data with import sets.
- Accepting external data onto the ECC queue.

Import and export properties page

You can use the Import Export Properties page to configure import and export properties.

To access the page, navigate to **System Properties > Import Export**.

The page lists properties categorized by either import or export, and by data format such as CSV or XML. The **Current Value** appears in black if a custom value is set, or in grey if no custom value is set.

Field normalization features

Field Normalization includes two features: normalization and transformation.

Normalization

Normalization forces the ServiceNow platform to convert different forms of the same field value to a single, accepted value automatically. By forcing a field to use a simple, recognizable description for multiple variations of the same thing, normalization can eliminate duplicate records and make searches easier. In addition to reconciling different forms of the same value in fields, normalization can be configured to adjust queries automatically to return normalized results.
Transformation

Transformation enables an administrator to transform raw field input into standardized values that are more meaningful to an organization. An example of a standardized value might be to round RAM size in configuration items to a whole number, such as 4000 MB instead of 4112 MB. Transformations are controlled by parameters and conditions and can be configured to return transformed values in queries.

Data management scripts

ServiceNow provides a robust platform for managing and manipulating data through scripts. The primary API for querying and manipulating data in tables within an instance is GlideRecord.

Data dictionary tables

These tables provide data dictionary, data modeling, and entity relationship information:

- **Tables** [sys_db_object]: contains a record for each table.
- **Dictionary Entries** [sys_dictionary]: contains additional details for each table and the definition for every column on each table. Each row represents either a column on a table or a table.
- **Field Labels** [sys_documentation]: contains the human-readable labels and language information.

Tables

The Tables [sys_db_object] table contains a record for each table in the database.

Access the Tables list by navigating to System Definition > Tables. Administrators can create a custom table, add or modify columns in a searchable and sortable embedded list, and define the auto-number format.

The following image shows a list of the tables that extend the Task table.
Dictionary Entries

The Dictionary Entries [sys_dictionary] table, also called the System Dictionary, defines every table and field in the system. It contains information about a field's data type, character limit, default value, dependency, and other attributes.

Access the system dictionary in one of these ways:

- To see the system dictionary list view, navigate to System Definition > Dictionary.
- To view particular dictionary definition, right-click the list header, form header, or field label, and select Configure Dictionary.

The following image shows a filtered list of dictionary entries for the Incident table and the Task table, which it extends.
Field Labels

The Field Labels [sys_documentation] table, also called the Language File, contains information about the labels and hints for each table and column in the system.

Access the language file in one of these ways:

- To see the list view, navigate to System Definition > Language File.
- To see the field label for a particular field, right-click the field label on the form.

The following image shows the language file filtered to display only labels on the Incident table.
The system dictionary is a table, called Dictionary Entry [sys_dictionary], that contains details for each table and the definition for every column on each table in an instance.

Each row in the system dictionary represents either a table or a column in one of the tables. The system dictionary provides options for administrators to modify tables and fields, which in turn define lists and forms.

Use caution when changing system dictionary records because changes can have a high impact on functionality. In particular, changes to dictionary entries for system tables, which are tables that begin with sys_, can create system-wide issues such as the inability to use update sets.

Dictionary changes are difficult to reverse. Also, dictionary changes automatically apply to all extended tables unless a dictionary override is defined. Be sure that changes are well-tested before applying them to a production instance.

Dictionary entry creation options
When you create a field from the system dictionary, it is automatically added at the end of the first section of the default form view.

In most cases, use the following interfaces rather than creating entries directly on the system dictionary:

• To create new tables and fields, use the Tables module.
• To create new fields, configure the table form.

**Dictionary overrides**

Dictionary overrides provide the ability to define a field on an extended table differently from the field on the parent table.

For example, for a field on the Task [task] table, a dictionary override can change the default value on the Incident [incident] table without affecting the default value on Task [task] or on Change [change].

Administrators can override these aspects of a field:

- Reference qualifiers
- Dictionary attributes
- Default values
- Calculations
- Field dependencies
- Default column display values
- Mandatory and read-only status

**Define a dictionary override**

Define dictionary overrides from a related list on the *dictionary entry* form for the field. You can add dictionary overrides only on tables that are in the same scope as the dictionary override.

1. Navigate to **System Definition** > **Dictionary**.
2. Open the record for the field.
3. In the **Dictionary Overrides** related list, click **New**.
4. Fill in the fields on the form, as appropriate (see table).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select the extended table to which the dictionary override applies.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The list shows only tables and database views that are in the same scope as the dictionary override.</td>
</tr>
<tr>
<td>Override reference qualifier</td>
<td>Select the check box to display the <strong>Reference qualifier</strong> field, which overrides the <strong>reference qualifier</strong> for the field on the extended table.</td>
</tr>
<tr>
<td>Override dependent</td>
<td>Select the check box to display the <strong>Dependent</strong> field, which overrides the field on which the current field depends.</td>
</tr>
<tr>
<td>Override attributes</td>
<td>Select the check box to display the <strong>Attributes</strong> field, which overrides the <strong>dictionary attributes</strong> for the field on the extended table.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Any attributes defined on the base table are ignored. If there are attributes on the base table that should still apply to the extended table, make sure to include them in this field.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Override default value</td>
<td>Select the check box to display the Default Value field, which overrides the default value for the field on the extended table.</td>
</tr>
<tr>
<td>Override calculation</td>
<td>Select the check box to display the Calculation field, which overrides the calculation of the value for the field on the extended table.</td>
</tr>
<tr>
<td>Override mandatory</td>
<td>Select the check box to display the Mandatory field, which overrides whether the field on the extended table must contain a value to save a record.</td>
</tr>
<tr>
<td>Override read only</td>
<td>Select the check box to display the Read only field, which overrides whether a user can change the field value on the extended table.</td>
</tr>
<tr>
<td>Override display value</td>
<td>Select the check box to use this field as the display value on the extended table. For example, the Story [rm_story] table uses the short description as the display value in reference fields instead of the number, as defined in the Task [task] table.</td>
</tr>
</tbody>
</table>

5. Click **Submit**.

### Modify a dictionary entry from a form
You can modify dictionary entries by configuring a field on a form.

1. Navigate to a field on a form.
2. Right click the field and select **Configure Dictionary** or **Show <field name>**.
   - The system dictionary entry for the field opens.
3. Update the dictionary entry fields.
4. Click **Update**.

### Modify Dictionary Entries from the Dictionary module
You can modify table and field dictionary entries from the Dictionary module.

1. Navigate to **System Definition > Dictionary**.
2. Click an entry for a field or table. Entries for tables have **Type** set to **Collection**.
3. Update the dictionary entry fields.
4. Click **Update**.

### Dictionary entry form
The Dictionary Entry form was redesigned to provide an Advanced view and additional fields. You might need to configure the form to see all fields.
### Table 322: Dictionary Entry Form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table</strong></td>
<td>Defines the table in which the element is created.</td>
</tr>
<tr>
<td>Note:</td>
<td>This list shows only the tables that meet the scope protections for adding fields. See.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Defines the field type of the column or, if the type is Collection, indicates that the dictionary entry represents the table.</td>
</tr>
<tr>
<td></td>
<td>You can change the type of a field. To preserve existing data, only change between logical types that map to the same physical type on the database. For example, Choice and String.</td>
</tr>
<tr>
<td><strong>Active</strong></td>
<td>Enables or disables the field. When this check box is cleared, the field is not used by the system.</td>
</tr>
<tr>
<td><strong>Read only</strong></td>
<td>Determines whether users can change the field value. When this check box is selected, users cannot change the value. The data for the field is calculated and displayed by the system.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can override this option for extended tables.</td>
</tr>
<tr>
<td><strong>Audit</strong></td>
<td>Enables or disables auditing for a table. Turning on Auditing (History) for a Table.</td>
</tr>
<tr>
<td>Note:</td>
<td>This option only applies to tables.</td>
</tr>
<tr>
<td><strong>Text index</strong></td>
<td>Determines whether searches index the text in a table.</td>
</tr>
<tr>
<td>Note:</td>
<td>This option only applies to tables. To exclude fields from indexing, see Remove an Index for a Specific Field.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Column label</td>
<td>Defines a unique label for the column. The label appears on list headers and form fields for the column.</td>
</tr>
<tr>
<td></td>
<td>• Updating the <strong>Column label</strong> field also updates the label in the language file (for the current language).</td>
</tr>
<tr>
<td></td>
<td>• When you create a new column, the column name is populated automatically based on the label, which is prefixed with u_ to indicate that it is custom. For example, if you enter <strong>Activity Description</strong> as the column label, the column name defaults to u_activity_description.</td>
</tr>
<tr>
<td>Column name</td>
<td>Defines the field name of the column. When you create a new field, this name is populated automatically based on the label and a prefix as follows:</td>
</tr>
<tr>
<td></td>
<td>• For a field on a table in a different scope, the name is prefixed with the scope to indicate that it is custom and not part of that application.</td>
</tr>
<tr>
<td></td>
<td>• For a field on a table in the same scoped application, the name does not have a prefix, which indicates that it is part of the application.</td>
</tr>
<tr>
<td></td>
<td>• For a field in a global application the name is prefixed with u_ to indicate that it is custom.</td>
</tr>
<tr>
<td></td>
<td>You cannot modify the prefix; however, you can modify the rest of the name. The name can contain only lowercase, alphanumeric ASCII characters and underscores (_). You cannot change the name of an existing dictionary record.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Max length | Provides a logical limit for the size of string fields to determine how the system displays them in the user interface and how to map them to physical database data types.  
String fields with a length under 255 characters appear as a single-line text fields. String fields with a length over 254 characters appear as a multi-line text box.  
The system maps the field length to the closest physical data type available on the database. In some cases this results in significantly more available length than originally specified. For example, entering a length of 50 maps to the closest physical data type of VARCHAR(100), which provides up to a 100 character limit or double the requested field length. Likewise, entering a length of 1000 maps to the closest physical data type of MEDIUMTEXT, which provides up to a 4000 character limit or four times the requested field length. |
| Note: | You can only change this value for a **String** field. Changes for any other type of field are ignored.  
Users on an Oracle instance cannot increase the maximum length of a string field to anything greater than 4000 through the application UI as this requires the CLOB datatype in Oracle. To increase beyond this size, log an incident with technical support to request the change.  
To prevent data from being lost, only decrease the length of a string field when you are developing a new application and not when a field contains data. A warning appears if a change to a custom field will result in data loss. For a base system field, you cannot make a change that will result in data loss. |
<p>| Mandatory | Determines whether this field must contain a value to save a record. For more information, see <strong>Creating Mandatory Fields</strong>.                                                                                          |
| Note: | You can override this option for extended tables.                                                                                                  |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>Indicates that this field is the display value for reference fields. Set this to true for the one field whose value you want to use as the text displayed in links to this table on lists and forms. By default, the Number field is the display value for all task tables.</td>
</tr>
</tbody>
</table>

**Note:**
- This option does not control whether a list or form displays this field as part of the layout. Instead, see List Configuration and Personalizing Forms.
- The display value becomes part of the form title when viewing an individual record from a table.
- You can set a different display value on an extended table than the display value on a parent table by using a dictionary override.

<table>
<thead>
<tr>
<th>Attributes [Advanced view]</th>
<th>Alters the behavior of a field or functionality that depends on the field. For more information, see Dictionary Attributes. Attributes can be overridden for extended tables with dictionary overrides. You can also configure attributes for this dictionary entry through the Attributes related list.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td></td>
</tr>
<tr>
<td>Use dynamic default [Advanced view]</td>
<td>Allows you to specify a default value that is generated dynamically based on a dynamic filter.</td>
</tr>
<tr>
<td>Dynamic filter value [Advanced view]</td>
<td>Specifies the dynamic filter that determines the default value if the Use dynamic default option is selected.</td>
</tr>
<tr>
<td>Default value</td>
<td>Specifies the default value of the field for any new record. Ensure that this value uses the correct field type. For example, an integer field uses a default value of 2 but cannot use a default value of two. These values can be overridden with dictionary overrides.</td>
</tr>
</tbody>
</table>

Reference Specification
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Makes the field into a reference field. If you enter a name that does not match an existing table, a new table is created when you save your changes to the dictionary record. If the current table has a module in the application navigator, a module for the new table is automatically created in the same application menu.</td>
</tr>
<tr>
<td>Use reference qualifier [Advanced view]</td>
<td>Specifies the type of qualifier to use:</td>
</tr>
<tr>
<td></td>
<td>• Simple: A set of choice lists where you can specify a reference qualifier condition.</td>
</tr>
<tr>
<td></td>
<td>• Dynamic: A dynamic filter that you can use to build the qualifier.</td>
</tr>
<tr>
<td></td>
<td>• Advanced: A static encoded query string or JavaScript code that you can use to build the qualifier.</td>
</tr>
<tr>
<td>Reference qual condition</td>
<td>Specifies a condition when the reference qualifier runs if the Simple qualifier type is selected.</td>
</tr>
<tr>
<td>Dynamic ref qual [Advanced view]</td>
<td>Specifies the dynamic filter that determines the reference qualifier when the Dynamic qualifier type is selected.</td>
</tr>
<tr>
<td>Reference qual [Advanced view]</td>
<td>Filters the records available for a reference field if the Advanced qualifier type is selected. Reference qualifiers can be overridden with dictionary overrides.</td>
</tr>
<tr>
<td>Reference key [Advanced view]</td>
<td>Identifies a field other than sys_id to use as the unique identifier for reference fields.</td>
</tr>
<tr>
<td>Reference cascade rule [Advanced view]</td>
<td>Defines what happens to a record if the record it references is deleted. Select one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Clear: clears the references (default).</td>
</tr>
<tr>
<td></td>
<td>• Delete: deletes all referencing records.</td>
</tr>
<tr>
<td></td>
<td>• Restrict: prevents record deletion if there is a referencing record.</td>
</tr>
<tr>
<td></td>
<td>• None: does not change referencing records.</td>
</tr>
<tr>
<td>Reference floats [Advanced view]</td>
<td>Enables the Edit button on related lists for one-to-many relationships.</td>
</tr>
<tr>
<td>Dynamic creation [Advanced view]</td>
<td>For reference fields, determines whether entering a value that does not match an existing record creates a new record on the referenced table. If selected, use the Dynamic creation script field to define how to create the new record.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dynamic creation script [Advanced view]</td>
<td>When the <strong>Dynamic creation field</strong> is selected, allows you to enter a script for creating a record on the referenced table.</td>
</tr>
<tr>
<td>Dependent Field</td>
<td></td>
</tr>
<tr>
<td>Dependent on field [Advanced view]</td>
<td>Specifies a field on which the current field depends. For more information, see Configuring Dependent Fields.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: You can override this value for extended tables.</td>
</tr>
<tr>
<td>Choice List Specification</td>
<td></td>
</tr>
<tr>
<td>Choice</td>
<td>Allows users to see a list of suggested values in one of the following ways:</td>
</tr>
<tr>
<td></td>
<td>• Drop-down menu without -- None --</td>
</tr>
<tr>
<td></td>
<td>• Drop-down menu with -- None --</td>
</tr>
<tr>
<td></td>
<td>• Suggestion</td>
</tr>
<tr>
<td></td>
<td>If a choice is used, either define a choice list or use the fields <strong>Choice table</strong> and <strong>Choice field</strong> to copy choices from another field elsewhere in the dictionary.</td>
</tr>
<tr>
<td>Choice table [Advanced view]</td>
<td>Populates the field choices with the same values as another choice field. If the <strong>Choice</strong> field is set to anything besides <strong>None</strong>, select a table to draw choice values from. The field <strong>Choice field</strong> must also be populated.</td>
</tr>
<tr>
<td></td>
<td>For example, if <strong>Choice table</strong> is set to the Incident [incident] table, this field has the same choice list as one of the choice fields on Incident.</td>
</tr>
<tr>
<td></td>
<td><strong>Choice field</strong> (see below) determines which field.</td>
</tr>
<tr>
<td>Choice field [Advanced view]</td>
<td>Populates the field choices with the same values as another choice field. If the <strong>Choice</strong> field is set to anything besides <strong>None</strong>, select a field from the table you selected for <strong>Choice table</strong>. For example, if the <strong>Choice table</strong> field is set to the Incident [incident] table, and <strong>Choice Field</strong> is set to <strong>Priority</strong>, this field has the same choices as the <strong>Priority</strong> field on Incident, even if those choices change.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This field must be a choice field.</td>
</tr>
<tr>
<td>Calculated Value</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Calculated [Advanced view]     | Determines whether the value of the field is calculated from other values. If selected, use the Calculation field to define how the calculation is performed. When sorting or grouping by a calculated field, the sort order is based on the field value from the last time the field was updated, not the last time the field was displayed.  

**Note:** In relation to business rules, calculated fields are populated first before any business rule, even a before business rule, is run. Calculated fields are then populated again if necessary after any before business rules run.  

| Calculation [Advanced view]    | When the Calculated field is selected, allows you to enter a script for calculating the value of the field. These can be overridden for extended tables with dictionary overrides. You can use the current object in this script. Just as with access control rules, the script can:  

• Evaluate to true or false.  
• Return an answer variable set to true or false.  
• Set a field value directly, such as:  
  
  `current.display_name="name"`.  

<table>
<thead>
<tr>
<th>Additional fields</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Identifies the table that the current table extends. Tables that do not extend other tables specify their own name in this field. For more information, see Tables and Classes.</td>
</tr>
<tr>
<td>Defaultsort</td>
<td>Obsolete. See Default list sort order.</td>
</tr>
<tr>
<td>Size class</td>
<td>Determines whether the platform handles this table as a large table by reducing the amount of memory stored for each row during queries. There is a scheduled job which runs and sets the value of this field.</td>
</tr>
<tr>
<td>Spell check</td>
<td>Enables or disables spell check on the field. For more information, see Adding Spell Check to a Field.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique</td>
<td>Requires the field value to be unique.</td>
</tr>
<tr>
<td>Warning:</td>
<td>Making a field unique when the corresponding table already has different</td>
</tr>
<tr>
<td></td>
<td>values for that field causes data loss. Before you enforce uniqueness on a</td>
</tr>
<tr>
<td></td>
<td>field, verify that no records in the table for the field have values, or</td>
</tr>
<tr>
<td></td>
<td>that they all have the same value.</td>
</tr>
</tbody>
</table>

### UI action

| Default view or Advanced view | Changes the form view to the default or advanced view. The fields change based on the view. If you are using the default view, you must write a script to accomplish the same tasks that advanced view fields provided. |

### Related Lists

| Access Controls | Provides access to the Access control rules on page 2498 that permit or limit access to the data in the table. |
| Choices         | Provides access to the options in the choice list field you are editing. |
| Dictionary overrides | Provides access to the dictionary overrides for this field. |
| Attributes      | Provides access to the dictionary attributes for this entry. |
| Labels          | Provides access to the labels used for the table or field you are editing. |

### Default list sort order

The system automatically sorts lists that meet one of the sort order criteria.

1. If there is a user preference specifying the sort order for the table, the system sorts by the field listed in the user preference.
2. If the table is the Task table or any of its extensions, the system sorts by the **Number** field.
3. If the table contains a field with the attribute isOrder=true, the system sorts by this field.
4. If the table contains an **Order** field, the system sorts by this field.
5. If the table contains an **Number** field, the system sorts by this field.
6. If the table contains an **Name** field, the system sorts by this field.

### Dictionary attributes

Dictionary attributes alter the behavior of the table or element that the dictionary record describes. Administrators can add or modify dictionary attributes.

#### Adding an Attribute
To add an attribute to a table or field, navigate to the **System Dictionary** record for the Dictionary entry, and add the attribute to the **Attributes** field. Attributes are comma-separated; if attributes already exist on a dictionary record, add a comma, with no spaces, before adding a new attribute.

For an attribute that accepts `true/false` values:

- To specify a value of **true**, you can either enter `attribute` or `attribute=true`.
- To specify a value of **false**, you can either ensure that the attribute does not appear or enter `attribute=false`. To maintain values during upgrades, do not remove an attribute that is on a table by default.

**Maintaining Attribute Values for Upgrades**

If you remove an attribute that is part of the base system, it is automatically restored during an upgrade. To prevent upgrades from changing the desired behavior of your system, leave the attribute on the table or field, but set its value as desired.

For example, if a field has the attribute `knowledge_search=true` by default, do not remove the attribute to set it to false; rather set it to `knowledge_search=false`.

**Available attributes**

This table lists the available dictionary attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Target Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_null</td>
<td>true/false</td>
<td>field_name field</td>
<td>If present or true, allows entering &quot;None&quot; as the field</td>
</tr>
<tr>
<td>allow_references</td>
<td>true/false</td>
<td>field_name field</td>
<td>If true, a tree is displayed to select from that includes reference fields so you can dot-walk.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>approval_user</td>
<td>name of field containing the user(s) for the approval type this field represents</td>
<td>integer field</td>
<td>The fields of the table are used to perform the lookup using a matcher. Approvals are specified as fields in the table that have an attribute of <code>approval_user=&lt;field_name&gt;</code>, where <code>&lt;field_name&gt;</code> indicates the field in the table that contains the users for this approval type. Fields with this attribute contain an integer value that indicates the sequence for the approvals. All approval fields with the same sequence number indicate that multiple approvals are required before continuing. Approvals are requested in the order of the sequence numbers. For example, all approvals with sequence number 100 must be approved before approvals with sequence number 200 are requested.</td>
</tr>
<tr>
<td>attachment_index</td>
<td>true/false</td>
<td>any table</td>
<td>If true, attachments on the table are indexed for search purposes. See <a href="#">Enable attachment indexing on a table</a> on page 1104.</td>
</tr>
<tr>
<td>barcode</td>
<td>true/false</td>
<td>string field</td>
<td>Allows a string field in the <a href="#">Mobile app UI</a> on page 1117 to access a mobile device's camera to scan and process a bar code.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>base_table</td>
<td>name of base table type</td>
<td>table_name</td>
<td>A table_name field allows the user to choose any table derived from the table specified by this attribute. By default, the base table itself is also included in the choice list (but see skip_root to turn off this behavior).</td>
</tr>
<tr>
<td>calendar_elements</td>
<td>list of field names, separated by semicolons (&quot;;&quot;)</td>
<td>any calendar event</td>
<td>Specifies a list of fields to be used when constructing the description of a calendar event. If not specified, the usual display name plus short description are used. The calendar_elements attribute does not support derived (dot-walked) fields.</td>
</tr>
<tr>
<td>collection_interval</td>
<td>interval specified as &quot;HH:MM:SS&quot; (like &quot;01:02:30&quot; for one hour, two minutes, and thirty seconds)</td>
<td>collection field</td>
<td>Specifies the interval of metrics collection.</td>
</tr>
<tr>
<td>close_states</td>
<td>inactive state integer values</td>
<td>task state</td>
<td>Used by the TaskStateUtil API - identifies the list inactive state values delimited by semicolons (;)</td>
</tr>
<tr>
<td>critical</td>
<td>true/false</td>
<td>any field in the apm_application table</td>
<td>Defines fields that are critical information about an application. This allows tracking the entry of critical information.</td>
</tr>
<tr>
<td>current_location</td>
<td>true/false</td>
<td>string field</td>
<td>Allows a string field in the Mobile app UI on page 1117 to access the GPS location of a mobile device.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>disable_execute_now</td>
<td>true/false</td>
<td>any table derived from sys_auto</td>
<td>If present or true, disables the usual <strong>Execute Now</strong> button. This is used by applications using schedules (such as Discovery) to substitute their own more appropriate action.</td>
</tr>
<tr>
<td>default_rows</td>
<td>integer value</td>
<td>mutitext fields</td>
<td>Sets the default number of rows in a mutitext field.</td>
</tr>
<tr>
<td>default_close_state</td>
<td>state integer value</td>
<td>task state field</td>
<td>Used by the TaskStateUtil API - identifies the default close state value for a task table</td>
</tr>
<tr>
<td>default_work_state</td>
<td>state integer value</td>
<td>task state field</td>
<td>Used by the TaskStateUtil API - identifies the default working state value for a task table</td>
</tr>
<tr>
<td>detail_row</td>
<td>name of field to display in detail row</td>
<td>any table</td>
<td>Displays the value of the specified field as a <strong>detail row</strong> for each record in the list view. UI15 is required to use this attribute.</td>
</tr>
<tr>
<td>email_client</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, causes an icon (an envelope) to appear in the more options menu in the form header. If clicked, a popup email client appears.</td>
</tr>
</tbody>
</table>

**Note:** If different **detail_row** attributes are defined for a parent table and a child table, the system uses the child table attribute.
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Target Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exclude_auto_recovery</td>
<td>true/false</td>
<td>any table</td>
<td>Disables automatic recovery of draft records for this table and its extensions.</td>
</tr>
<tr>
<td>extensions_only</td>
<td>true/false</td>
<td>any table</td>
<td>Table should only have records in tables that extend it. For example, the Task table has this attribute because you would create incident, problem, change records and not task records.</td>
</tr>
<tr>
<td>field_list_selector</td>
<td>true/false</td>
<td>any glide_list</td>
<td>Allows the user to select a field from the dependent table (or current if dependent is not specified). This is used in some workflow activities.</td>
</tr>
<tr>
<td>field_decorations</td>
<td>UI Macro name list, separated by semicolons (&quot;;&quot;)</td>
<td>most fields (except multi-line text fields)</td>
<td>Similar to ref_contributions, causes the named UI macro to be invoked when the field is rendered on a form.</td>
</tr>
<tr>
<td>format</td>
<td>format name</td>
<td>any numeric field</td>
<td>Specifies a named format to use instead of the standard numeric formatting. Options are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• glide_duration: formats a time specified in milliseconds as ddd hh:mm:ss.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• none: disables automatic number formatting (for example, changes 2,500 to 2500).</td>
</tr>
<tr>
<td>fv</td>
<td>table name; field name; sys_id</td>
<td>field_value field</td>
<td>This uses the three values to set the display of the field_value field.</td>
</tr>
<tr>
<td>glide.db.oracle.ps.query</td>
<td>true/false</td>
<td>any table</td>
<td>If present and false, prevents the use of Oracle prepared queries on the table.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>global_visibility</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, makes this table visible globally even if there are domain restrictions (that is, the sys_domain field has a value).</td>
</tr>
<tr>
<td>hasLabels</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, marks this table as being the target of a label at some point. This attribute can be set manually, but it is set automatically whenever a label is generated. When true, the label engine will run on any change to the table, updating the labels as needed.</td>
</tr>
<tr>
<td>hasListeners</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, marks this table as available for listeners to get events (insert, update, delete) on.</td>
</tr>
<tr>
<td>hasWorkflow</td>
<td>true/false</td>
<td>any table</td>
<td>Tells the workflow engine to listen for changes to the table, firing events to a workflow when a record associated with a particular workflow has changed.</td>
</tr>
<tr>
<td>html_sanitize</td>
<td>true/false</td>
<td>any field</td>
<td>If present or true, HTML sanitization is enabled for the selected field.</td>
</tr>
<tr>
<td>icons</td>
<td>name of JavaScript class</td>
<td>any workflow field</td>
<td>Specifies a JavaScript class that produces workflow icons.</td>
</tr>
<tr>
<td>image</td>
<td>relative path of image file</td>
<td>any table</td>
<td>Specifies an image file to be used when the table is used in a module or BSM map. This specification overrides the icons that would otherwise be used for the table.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td>-------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>include_container_types</td>
<td>true/false</td>
<td>any internal_type field</td>
<td>Causes the field to render with container (split) types as well other types.</td>
</tr>
<tr>
<td>iterativeDelete</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, forces all row deletes to be executed iteratively. Otherwise, some deletes may be performed using a more efficient bulk method.</td>
</tr>
<tr>
<td>knowledge_custom</td>
<td>name of JavaScript function</td>
<td>any field</td>
<td>Specifies a JavaScript function to implement a custom knowledge search (see knowledge_search).</td>
</tr>
<tr>
<td>knowledge_search</td>
<td>true/false</td>
<td>string fields</td>
<td>If present or true, causes a knowledge search icon (a small book) to appear next to the field. Clicking this icon launches a pop-up window for searching the knowledge base, unless a custom knowledge search function has been specified (see knowledge_custom).</td>
</tr>
<tr>
<td>largeTable</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, marks this table as &quot;large&quot; for the purpose of preventing table locking with specific MySQL database operations (adding/removing a column/index, compacting a table). Without this attribute (or the smallTable attribute), whether a table is large is determined by the glide.db.large.threshold property, or the default value of 5,000.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>listen</td>
<td>true/false</td>
<td>any field</td>
<td>If present or true, causes a call to a JavaScript function named <code>&lt;tableName&gt;_&lt;fieldName&gt;Listen</code> or <code>globalListen</code> if that function does not exist. The function is called with arguments (tableName, fieldName, oldValue, newValue).</td>
</tr>
<tr>
<td>live_feed</td>
<td>true/false</td>
<td>any field</td>
<td>If present or true, creates a toggle option on the activity formatter header for incidents, tasks, and problems. The toggle provides the choice between the <strong>Live Feed</strong> for that record (also known as a document feed) or the activity formatter fields already in use. See Activity formatter on page 764 for more details.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>long_label</td>
<td>true/false</td>
<td>any field</td>
<td>Long or short labels refer to the label that is displayed for reference fields on a form. For example, if the field contains the caller’s email address, the long label would be Caller Email while the short label would just be Email. Usually the placement of the field on the form makes it clear what the field represents. The global property (glide.short.labels) is used to specify the type of labels that are displayed for all reference fields on any form. This global property can be overridden for any field by setting the short_label=true or long_label=true attribute for the field in the Dictionary.</td>
</tr>
<tr>
<td>mode_toggler</td>
<td>true/false</td>
<td>any composite_name field</td>
<td>If present or true, causes a name mode toggle icon (a small right-pointing triangle) to appear to the right of the label. Clicking this icon causes the field's rendering to change from a text field accepting <code>&lt;tablename&gt;.&lt;fieldname&gt;</code> to a pair of reference choice boxes (one for the table, the other for the field). The latter is the default.</td>
</tr>
<tr>
<td>model_class</td>
<td>binary Java class name</td>
<td>any field of type glide_var</td>
<td>Specifies a model variable within Java code. The model must have a class that implements the IVariablesModel interface.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>model_field</td>
<td>see description</td>
<td>any field of type glide_var</td>
<td>Identifies a reference field in the record that has the model defined for it. For example, a workflow activity is associated with an activity definition. The activity definition has a related list of questions that make up the model for that activity definition. By using the activity_definition as the model_field for the activity, the model for the workflow activity is built by reading the questions that are defined for the referenced activity definition.</td>
</tr>
<tr>
<td>nibble_size</td>
<td>positive integer</td>
<td>any table affected by the table cleaner.</td>
<td>Specifies the maximum number of records the table cleaner can delete in a single operation. The default value for this attribute is 250.</td>
</tr>
<tr>
<td>nibble_sleep</td>
<td>true/false</td>
<td>any table affected by the table cleaner.</td>
<td>If false, causes the table cleaner to perform cleanup operations without a pause between each operation.</td>
</tr>
<tr>
<td>no_attachment</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, prevents the attachment icon (a paperclip) from appearing on the form header.</td>
</tr>
<tr>
<td>no_attachments</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, attachments will not be checked for and deleted when a record from this table is deleted. Meant for high-activity tables that never have attachments.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>no_audit</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, this field will not be audited, even if the table is being audited.</td>
</tr>
<tr>
<td>no_audit_delete</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, a sys_audit_delete record will never be created when a record from this table is deleted. Meant for high-activity tables that never need sys_audit_delete information.</td>
</tr>
<tr>
<td>no_auto_map</td>
<td>true/false</td>
<td>any table</td>
<td>If true, this field will not be mapped during an import set. This is primarily used for LDAP imports.</td>
</tr>
<tr>
<td>no_email</td>
<td>true/false</td>
<td>any glide_list field referencing sys_user</td>
<td>If present or true, the email box is removed from the glide_list field like the Watch list field.</td>
</tr>
<tr>
<td>no_multiple</td>
<td>true/false</td>
<td>any glide_list field</td>
<td>Hides the select multiple icon.</td>
</tr>
<tr>
<td>no_optimize</td>
<td>true/false</td>
<td>any table affected by the table cleaner.</td>
<td>If present or true, prevents the MySQL table compaction operation from running on the specified table. The table compaction operation normally runs after the table cleaner deletes at least 50% of the data in the specified table.</td>
</tr>
<tr>
<td>no_separation</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, marks this table as not participating in domain separation.</td>
</tr>
<tr>
<td>no_text_index</td>
<td>true/false</td>
<td>any field on a text indexed table</td>
<td>If a table is text indexed, the no_text_index attribute on a field will prevent this field from being included in the text index.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>no_truncate</td>
<td>true/false</td>
<td>any string field</td>
<td>In a list view, shows the entire text value of the multi-text value in a list, without truncating it. Without this attribute the string is truncated based on the UI property <strong>Number of characters displayed in list cells</strong> which is 40 by default.</td>
</tr>
<tr>
<td>no_update</td>
<td>true/false</td>
<td>table</td>
<td>Is true for tables in which records are inserted or deleted but not updated. Prevents the system from creating sys_mod_count, sys_updated_by, sys_updated_on fields in the table when it is created. Does not stop the table from being updated. This attribute is used to save space on high volume system tables, such as syslog and sys_audit.</td>
</tr>
<tr>
<td>no_view</td>
<td>true/false</td>
<td>any glide_list field</td>
<td>Hides the view selected item icon.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>onlineAlter</td>
<td>true/false</td>
<td>any table</td>
<td>Tables with the onlineAlter attribute perform MySQL database operations using online schema changes. Online schema changes provides a lock-free table upgrade when adding, modifying, or removing columns and when adding or dropping indexes. Without online schema changes, these changes to the database lock write access during execution. Online schema changes use additional system resources. Oracle databases do not lock tables by default and do not use online schema changes.</td>
</tr>
<tr>
<td>order</td>
<td>numeric value</td>
<td>model variable fields</td>
<td>Used internally only (for model variables).</td>
</tr>
<tr>
<td>popup_processor</td>
<td>binary Java class name</td>
<td>any field or table</td>
<td>Specifies a custom popup processor for processing the field (or all fields in a table).</td>
</tr>
<tr>
<td>readable</td>
<td>true/false</td>
<td>any conditions field</td>
<td>When true, causes the conditions field to be rendered in any list view as a human-readable condition (instead of the encoded query actually stored in the database). The form view for this field is unaffected.</td>
</tr>
<tr>
<td>ref_ac_columns</td>
<td>list of field names separated by semicolons</td>
<td>any reference field with an auto completer (see ref_auto_completer)</td>
<td>Specifies the columns whose display values should appear in an auto completion list in addition to the name. See the cmdb_ci field (Configuration Item) on the Incident form for a working example.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ref_ac_columns_search</td>
<td>true/false</td>
<td>any reference field with an auto completer (see ref_auto_completer)</td>
<td>Causes auto-complete to work with all fields specified in the ref_ac_columns attribute. This overrides the default behavior, which searches only the display value column. See Configure auto-complete to match text from any reference field on page 882.</td>
</tr>
<tr>
<td>ref_ac_display_value</td>
<td>true/false</td>
<td>any reference field with an auto completer (see ref_auto_completer)</td>
<td>Causes the reference field to hide a the display value column so that auto-complete only matches text from the columns listed in the ref_ac_columns attribute. This feature requires the use of the AJAXTableCompleter class and the ref_ac_columns, ref_ac_columns_search, and ref_ac_display_value attributes. See Remove the display value column on page 880.</td>
</tr>
<tr>
<td>ref_ac_order_by</td>
<td>field name</td>
<td>any reference field with an auto completer (see ref_auto_completer)</td>
<td>Specifies the column that will be used to order the auto completion list.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Target Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| ref_auto_completer    | JavaScript class name              | any reference field (can be applied to a table to affect all reference fields on the table.) | Specifies the name of a JavaScript class (client side) that creates the drop-down auto completion choices. Valid class values include:  
  - **AJAXReferenceCompleter**: Displays matching auto-complete choices as a drop-down choice-list. The list only displays the reference table's display value column. Reference fields automatically use this class if there is no other auto-completion class specified.  
  - **AJAXTableCompleter**: Displays matching auto-complete choices as rows in a table. The table displays the reference table's display value column and any columns listed in the ref_ac_columns attribute.  
  - **AJAXReferenceChoice**: Displays matching auto-complete choices as a drop-down choice-list. The list only displays the reference table's display value column. Furthermore, the list only displays up to 25 matching choices. If there are more than 25 auto-complete choices, the reference field instead displays the choices with the AJAXTableCompleter class.  

For more information, see Auto-complete for reference fields on page 877.
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Target Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref_contributions</td>
<td>UI Macro name list, separated by semicolons (&quot;;&quot;))</td>
<td>any reference field</td>
<td>Causes the named UI macro to be invoked when the field is rendered on a form.</td>
</tr>
<tr>
<td>ref_list_label</td>
<td>label text</td>
<td>any table</td>
<td>Specifies the title to use in a list banner.</td>
</tr>
<tr>
<td>ref_qual_elements</td>
<td>field name list, separated by semicolons (&quot;;&quot;))</td>
<td>any reference field with a reference_qual field</td>
<td>Specifies a list of fields to be sent back to the server in order to get an updated reference.</td>
</tr>
<tr>
<td>ref_sequence</td>
<td>list of fields in referenced table, separated by top hats (&quot;^^&quot;)</td>
<td>any reference field</td>
<td>Specifies the fields in the referenced table that should be used to order the choice list. This works like an ORDER BY clause in SQL, with each element in ascending order.</td>
</tr>
<tr>
<td>reference_types</td>
<td>list of valid reference types that are clickable separated by semicolons (&quot;;&quot;))</td>
<td>field_name field</td>
<td>Limits the reference fields that are displayed in the tree to the specified types.</td>
</tr>
<tr>
<td>remoteDependent</td>
<td>name of database and table (like &quot;model.matcher&quot;)</td>
<td>any script field</td>
<td>Defines the remote (such as, in another database) table that the script depends on.</td>
</tr>
<tr>
<td>repeat_type_field</td>
<td>field name</td>
<td>a repeat count field for schedule rotation</td>
<td>Specifies the field that contains the repeat type (daily, weekly, monthly, or yearly).</td>
</tr>
<tr>
<td>restrictTo</td>
<td>field name (including indirect, dot-walked field references)</td>
<td>any conditions field</td>
<td>Specifies the field that contains the comma-separated list of fields that the conditions should be restricted to using.</td>
</tr>
<tr>
<td>ro_collapsible</td>
<td>true/false</td>
<td>any multi-line field</td>
<td>If present or true, causes an icon (either a &quot;+&quot; or a &quot;+&quot;) to appear next to the field's label, allowing the field itself to be expanded or collapsed.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>scale</td>
<td>integer</td>
<td>decimal field</td>
<td>Sets the number of decimal places to use on the Decimal field type. The default is 2. This is applied to the Max Length of the field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Increase the Max Length to a value greater than 15 to increase this attribute.</td>
</tr>
<tr>
<td>script</td>
<td>a function that returns the contents of the field</td>
<td>any slushbucket field</td>
<td>Allows you to write a script to define what will be loaded into the slushbucket field.</td>
</tr>
<tr>
<td>short_label</td>
<td>true/false</td>
<td>any field</td>
<td>Long or short labels refer to the label that is displayed for reference fields on a form. For example, if the field contains the caller’s email address, the long label would be Caller Email while the short label would just be Email. Usually the placement of the field on the form makes it clear what the field represents. The global property (glide.short.labels) is used to specify the type of labels that are displayed for all reference fields on any form. This global property can be overridden for any field by setting the short_label=true or long_label=true attribute for the field in the dictionary.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------</td>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>show_all_tables</td>
<td>true/false</td>
<td>document ID fields</td>
<td>Allows users to select documents from system tables. For example, sys_script or sys_user. By default, users cannot select records from system tables.</td>
</tr>
<tr>
<td>show_condition_count</td>
<td>true/false</td>
<td>condition fields</td>
<td>Enables or disables the condition count widget to preview how many records would be returned by a set of conditions.</td>
</tr>
<tr>
<td>skip_root</td>
<td>true/false</td>
<td>table_name field</td>
<td>If present or true, removes the base table from the choice list (see base_table for more details).</td>
</tr>
<tr>
<td>sla_basis</td>
<td>list of table names separated by semicolons (&quot;,&quot;;)</td>
<td>any field of date type (glide_date_time, glide_date, due_date, date, or datetime)</td>
<td>Defines the tables for which this field determines the start (open) time of an SLA.</td>
</tr>
<tr>
<td>sla_closure</td>
<td>list of table names separated by semicolons (&quot;,&quot;;)</td>
<td>any field of date type (glide_date_time, glide_date, due_date, date, or datetime)</td>
<td>Defines the tables for which this field determines the start (open) time of an SLA.</td>
</tr>
<tr>
<td>slushbucket_ref_no_expand</td>
<td>true/false</td>
<td>any reference field</td>
<td>If present or true, prevents users from expanding the field from a form or list slushbucket.</td>
</tr>
<tr>
<td>smallTable</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, marks this table as &quot;small&quot; (that is, not large) for the purposes of our querying strategy. Without this attribute (or the largeTable attribute), whether a table is large is determined by the glide.db.large.threshold property, or the default value of 5,000.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>start_locked</td>
<td>true/false</td>
<td>any glide_list field</td>
<td>Determines whether the field is locked or unlocked by default. Set the value to false to unlock the field by default.</td>
</tr>
<tr>
<td>staticDependent</td>
<td>name of table</td>
<td>any script field</td>
<td>Defines the table that the script depends on.</td>
</tr>
<tr>
<td>strip_html_in_pdf</td>
<td>true/false</td>
<td>any field</td>
<td>Attempts to remove HTML tags from a field when that field is exported to a PDF. Most likely useful on HTML fields.</td>
</tr>
<tr>
<td>synch_attachments</td>
<td>true/false</td>
<td>any table</td>
<td>Similar to update_synch but writes the record's file attachments to update sets. See Enable attachment indexing on a table on page 1104.</td>
</tr>
<tr>
<td>table</td>
<td>name of table</td>
<td>field_name field</td>
<td>Displays the fields of the table specified.</td>
</tr>
<tr>
<td>tableChoicesScript</td>
<td>name of script include</td>
<td>table_name field</td>
<td>The name of a script include whose process() method returns an array of table names from which to select.</td>
</tr>
<tr>
<td>target_form</td>
<td>name of form</td>
<td>any table</td>
<td>Specifies the alternative form to be used when this table is referenced through a popup on a reference field.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>text_index_filter_junk</td>
<td>true/false</td>
<td>any table</td>
<td>Set the value to <strong>false</strong> to disable the junk filter for the table. By default, Zing does not index or search for 2-digit numbers and single character words (unless they are Chinese or Japanese characters). You must regenerate the index after disabling the junk filter. This attribute results in a larger table index. For optimal performance, do not apply it unless it is required.</td>
</tr>
<tr>
<td>text_search_only</td>
<td>true/false</td>
<td>table_name field</td>
<td>Limits the tables listed to those that are searchable by text.</td>
</tr>
<tr>
<td>text_index_translation</td>
<td>true/false</td>
<td>any table</td>
<td>If present or true, forces indexes to be recalculated when translated strings are added. Requires sys-admin role to modify. Automatically set for indexed fields that are translated, and to fields that have a translation and are being indexed. This attribute is overridden by the glide.i18n.force_index system property, which defaults to true.</td>
</tr>
<tr>
<td>time_zone_field</td>
<td>name of field containing the time zone</td>
<td>any schedule date/time field</td>
<td>Specifies the field in the parent record that contains the reference time zone for this field.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>---------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timeDimension</td>
<td>true/false</td>
<td>any field of date type (glide_date_time, glide_date, due_date, date, or datetime) in a table subclassed from the task table</td>
<td>If present or true, enables production of time dimension data for use by OLAP (to produce reports based on quarters, weeks, or other time periods). Note: OLAP functionality has been deprecated.</td>
</tr>
<tr>
<td>tree_picker</td>
<td>true/false</td>
<td>reference field with reference to a hierarchical table</td>
<td>Displays the hierarchy of reference values in a tree display (such as locations).</td>
</tr>
<tr>
<td>ts_weight</td>
<td>integer value</td>
<td>any field</td>
<td>Controls the relative importance of a match in the field for text search. See Control Match Relevance By Field.</td>
</tr>
<tr>
<td>types</td>
<td>list of valid element types separated by semicolons (&quot;,&quot;), field_name field</td>
<td>Limits the fields display to the specified types.</td>
<td></td>
</tr>
<tr>
<td>update_exempt</td>
<td>true/false</td>
<td>field on any table where update_synch=true</td>
<td>If present or true, you can change this field without skipping updates to the rest of the record. During software upgrades, the value of this field is preserved, while the rest of the record receives upgrades. By default, the Active field on a tracked table is treated as update_exempt even if the attribute is not present.</td>
</tr>
<tr>
<td>update_synch</td>
<td>true/false</td>
<td>any table</td>
<td>Indicates that changes in the table are tracked in update sets. Administrators cannot modify this attribute. To migrate data, use an instance-to-instance import.</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target Element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>update_synch_custom</td>
<td>binary Java class name</td>
<td>any table</td>
<td>Specifies custom update producer (a Java class) that handles update set production for this table.</td>
</tr>
<tr>
<td>use_workflow</td>
<td>true/false</td>
<td>any table that has delivery plans or uses workflow</td>
<td>If present or true, causes workflow to be used instead of delivery plans.</td>
</tr>
<tr>
<td>user_preference</td>
<td>true/false</td>
<td>any field</td>
<td>If present or true, causes any user preferences to be used instead of the normal default value.</td>
</tr>
</tbody>
</table>

Modify the Glide durations format

In order to convert fields that are displayed in milliseconds (such as ‘Resolution Time’ on the Incident table) to a duration format (Months/Days/Hours/Seconds), populate the attribute field on the dictionary with: `format=glide_duration`.

Now the value will be displayed as a duration in lists and can be utilized in the Configure Calculations functions. This may be very useful to illustrate ITIL KPIs such as Mean Time To Repair.

Field Dictionary:
Figure 420: Duration Dict

Results in list:
This page describes the approaches to managing table size growth and archiving old data.

With data constantly being added to the system, and activity being logged into system tables in the database, these tables grow in size and require management. As data sets increase in size, the amount of I/O traffic associated with actions such as cleaning, deleting, and archiving can negatively affect the performance of an instance. Additionally, working with all rows in a data set, rather than a smaller working set, can create unnecessary risk.

The Database Rotation plugin preserves instance performance and averts risk associated with querying growing data sets utilizing two techniques. Both techniques are based on the concept of managing large quantities of data by separating whole sets into individual tables based on user-specified time parameters. After this task is performed, each technique handles data in a different manner:

- **Table Rotation** works by rotating among a small set of tables, and deleting and reusing the old tables for new data.
- **Table Extension** works by periodically starting a new table and allowing old tables to be easily archived and removed from the system.

For an additional archiving option see [Archiving Data](#).

**Included with Database Rotation plugins**

The Data Management and Archiving techniques are contained in two separate plugins.

- **Database Rotations Plugin** activates Table Rotation and Extension without any tables automatically included (com.snc.db.rotation)
- **Database Rotations Default Tables Plugin** applies Table Rotation and Extension to specific tables (com.snc.db.rotation_default_tables)
### Table 324: Database Rotation

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Tables applied to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Rotation</td>
<td>syslog</td>
</tr>
<tr>
<td></td>
<td>sys_querystat</td>
</tr>
<tr>
<td></td>
<td>ecc_queue</td>
</tr>
<tr>
<td></td>
<td>ecc_event</td>
</tr>
<tr>
<td></td>
<td>cmdb_metric</td>
</tr>
<tr>
<td></td>
<td>sysevent</td>
</tr>
<tr>
<td>Table Extension</td>
<td>sys_audit</td>
</tr>
<tr>
<td></td>
<td>sys_email</td>
</tr>
</tbody>
</table>

**Activate database rotation**

For new instances, database rotation is active by default. If you are upgrading from a previous version, you can activate the Database Rotations plugin if it is not already active.

**Note:** Deployment of this plugin should be executed in partnership with a ServiceNow representative.

If it is not already active, you can activate the Database Rotations plugin if you have the admin role.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Do not activate the Database Rotations Default Tables plugin. Instead, specify the tables manually, after consulting a ServiceNow representative.

**Archive data**

The archive application moves data that is no longer necessary for immediate day-to-day access from primary tables into a set of archive tables.

The longer an instance runs, the more likely it is to accumulate data that is no longer relevant to current business needs. For example, task records from two years ago are typically less relevant than currently active tasks. Old data may eventually cause performance issues by consuming system resources and slowing down queries and reports. If you cannot delete this data because you need it for auditing or for historical purposes, archive the data to remove it from immediate access and free up system resources.

Data archiving supports domain separation. For example, incidents that belong to a domain keep their domain designation even after they are archived.
Figure 422: Sample Benefits of Archiving Data

To setup data archiving:

1. Activate the Data Archiving plugin.
2. Create an archive rule for the table you want to archive.
3. Define how the archive rule handles related records.
4. Verify how many records the archive rule affects.
5. Activate the archive rule.

Activate data archiving

The Data Archiving plugin installs the System Archiving application.
Figure 423: System Archiving Application

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

*Create an archive rule*

The System Archiving application includes several sample archive rules that illustrate the archive features.

**Note:** The sample archive rules are not intended as best practice and are inactive by default.

1. Navigate to **System Archiving > Archive Rules**.
2. Click **New**.
3. Fill in the fields as appropriate.

**Table 325: Archive Rule form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Required. Enter a unique name that identifies the rule. Since this is the display field for archive rules, references to archive rules display the name you enter here.</td>
</tr>
<tr>
<td>Table</td>
<td>Required. Select the table containing records to archive. There can only be one archive rule per table. If there is an existing archive rule for a table, the table no longer appears as an option on the archive rule table list.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to enable the archive rule. Clear the check box to disable the rule. ServiceNow recommends leaving your archive rules inactive until you calculate an estimate of the number of records the rule affects and verify that rule behaves as expected.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description of the archive rule.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditions</td>
<td>Required. Select the fields and values that must be true in order for the archive rule to run. Typically, you would archive inactive records older than a certain date.</td>
</tr>
<tr>
<td>Parent</td>
<td>Select an existing archive rule on which this rule depends to run. This archive rule only runs when the parent rule also runs.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

**Activate the archive rule**

After verifying that the archive rule is selecting records as expected, activate the archive rule.

The instance runs the archive rule as a *schedule job* set to run every 60 minutes.

1. Navigate to **System Archiving > Archive Rules**.
2. Select the archive rule you want to activate.
3. Select the **Active** check box.
4. Click **Update**.

**Activate the archive rule immediately**

If you do not want to wait for the scheduled job to run the archive rule, you can manually start the archive rule.

1. Navigate to **System Archiving > Archive Rules**.
2. Select the archive rule you want to run.
3. Click the **Run Archive Now** related link.

**Note:** The archive rule must be active in order to see the link.

**Set archive rule processing behavior**

In order to prevent the archive process from consuming too many system resources, the instance uses several system properties to control how many records the archive rule processes at one interval.

By default an archive rule follows these processing rules:

- Archives 100 records for each batch job
- Sleeps 1 second between batch jobs
- Runs 10 batch jobs in an archive run (every hour)

You can change these default settings by *adding the following system properties*.

**Table 326: Archive System Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.db.archive.batch_size</td>
<td>Controls how many records an archive rule processes per batch job.</td>
<td>Integer</td>
<td>100</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Default Value</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>glide.db.archive.sleep_time</td>
<td>Controls how long each archive rule batch job runs, in seconds.</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>glide.db.archive.max_iterations</td>
<td>Controls the maximum number of batch jobs to run within an hour.</td>
<td>Integer</td>
<td>10</td>
</tr>
</tbody>
</table>

**Tables and modules created by an archive rule**

The first time you activate an archive rule, it performs the following actions:

- Creates the archive table in the database. The archive table has the same name as the primary table with an "ar_" prefix. For example, if you archive the Incident [incident] table, then the archive table is [ar_incident].
- Converts multiple joined tables into a single flat-file archive table. The archive table no longer consists of a base and extended tables.
- Converts reference field values (values set by references to records in other tables) into string values. The archive record contains the display value of the reference field at the time of the archive.
- Adds a module to the **Archive Tables** list in the **System Archiving** application. The module name is a combination of the word "Archive" plus the display name for the archived table. For example, the archive module for the Attachment [sys_attachments] table is **Archive Attachment**. Click the module name to view records in the archive table.
- Creates a list of the archive table using the default list view.
- Creates a form for the archive table using the default form view. The form excludes any **dot-walking** fields such as **Caller ID.Email**.
- Creates an activity formatter and variable formatter if they exist for the table being archived.
Figure 424: Conversion of Multiple Joined Tables into a Flat Archive Table

**Change an archive schedule**
All active archive rules are executed by a system scheduled job set to run every 60 minutes. You can modify the job if you need to change the interval.

1. Navigate to **System Scheduler > Scheduled Jobs**.
2. Open the **Archive** record.
3. Modify the **repeat** value.

**Query archived tables**
Archived tables are not optimized for ad hoc queries. They only contain index entries for the display value, creation date, and the primary key of sys_id.
For this reason ServiceNow does not recommend making ad hoc queries against an archived table, such as searching for all priority 1 archived incidents. Instead, only search against the indexed fields. For example, search for incident INC100001 or incidents created on a specific date.

**Setting the language of archived strings**

On internationalized instances, the archive process uses the language of the SYSTEM user to select the display value strings.

If there is no SYSTEM user, the instance uses the default language setting to select the display value strings. You can either create a SYSTEM user with a specific language setting or set the system default language to select the language of archived strings.

---

**Figure 425: Sample Archive Displayed in French**

*Verify the number of records affected*

Each archive rule provides an estimate of the number of records the rule affects in the Record estimate field.

This estimate only includes primary records and excludes any related records added to the archive rule. The estimate helps you determine if the archive rule affects the number of records you expect it to. If the estimate is too high or low, change the archive rule conditions and then click the Recalculate Estimate related link.

1. Navigate to System Archiving > Archive Rules.
2. Select the archive rule you want to estimate records for.
3. Click the Recalculate Estimate related link.

*Archive related records*

Use the Archive Related Records related list to add related records to the archive rule.

1. Navigate to System Archiving > Archive Rules.
2. Select the appropriate archive rule.
3. From the **Archive Related Records** related list, click **New**.

4. Fill in the Archive Related Records form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive map</td>
<td>Displays the archive rule to which the related records apply.</td>
</tr>
<tr>
<td>Action</td>
<td>Select the action you want the archive rule to take on related records.</td>
</tr>
<tr>
<td></td>
<td>Choices include:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Archive</strong>: archive records that reference the archived record.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Clear</strong>: remove the reference to the archived record.</td>
</tr>
<tr>
<td></td>
<td>The record no longer references the archived record and does not appear as</td>
</tr>
<tr>
<td></td>
<td>the related record in future archives.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Delete</strong>: delete records that reference the archived record.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Reference</td>
<td>Select the relationship of the records you want to apply an action to. The Reference field lists all relations that currently exist for the table being referenced. There are two types of possible relations.</td>
</tr>
</tbody>
</table>

- Another table has a reference field calling the archived table. For example, if you are archiving problem records, there is a Problem ID field in the Incident table that references the related problem records.

- The Archive action archives the related record in addition to the primary record. For example, if you select the Problem in Incident reference, the related record rule also archives any incident record that references an archived problem.

- The Clear action removes the reference to the primary record. For example, if you select the Problem in Incident reference, the related record rule updates any incident record with a reference to the archived problem record by clearing the reference. If the reference is a many-to-many relationship, the related record rule deletes the reference instead of clearing the reference.

- The Delete action deletes any record that references the primary record. For example, if you select the Problem in Incident reference, the related record rule deletes any incident record that references the archived problem record.

- Another table has a Document ID field which might point to the archived table. For example, if you are archiving problem records, there is a sys ID field in the Attachments table that may reference the problem record. The list indicates document ID relationships by displaying an asterisk (*) character at the end of the selection name.

- The Archive action updates the Document ID of the related record to point to the archived table. For example, if you select the Table sys ID Attachment(sys_attachment)* reference, the related record rule updates the attachment record to change the Document ID to refer to...
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference table</td>
<td>Displays the table where the rule looks for related records.</td>
</tr>
<tr>
<td>Reference element</td>
<td>Displays the reference field or <strong>Document ID</strong> the rule queries for.</td>
</tr>
<tr>
<td>Reference table rule</td>
<td>Select an existing archive rule that applies to the related records you are archiving. For example, if you already have an archive rule for the Incident table, you can select the existing Incident table rule when archiving records related to incidents records.</td>
</tr>
</tbody>
</table>

5. Click **Submit**.

**Archive a restored record**

Archive rules will not archive restored records. In order to archive a restored record you must manually archive the record.

1. Navigate to the restored record.
2. Click the **Archive Record** link.
View archived data
You can view archived tables from the **Archived Tables** list.

1. Navigate to **System Archiving > Archive Tables**.

2. Click the module name for the archived table you want to view. For example, **Archive Attachment**.
3. Click a record from the archived table to see the archived record.
Restore archived data
When you restore a record, the instance inserts it back into the primary table and flags the record as having been restored in the log.

1. Navigate to System Archiving > Archive Log.
2. Select the archived record to restore.
3. Click the Restore Record related link.
Reference values converted to strings

Archived data is stored as a flat file with no reference fields to other tables. The archive process converts any references to other tables to string values.

In the case of a reference field, the string uses the display value such as the caller’s user name. For example, the Caller reference field in an incident would display the string ITIL User. If the reference was a document ID and the archive rule included the option to archive related document IDs, then the string is the document ID of the related record.

It is important to note that archive records do not receive any future changes to referenced values. For example, if you change the user name for "John Smith" to "John A Smith", all active incident records automatically show the caller as "John A Smith" because of the reference between the Incident and User tables. However, all archived incident records display the user name that existed at the time of the archive.

Any incident for "John Smith" continues referencing this user. Likewise, if you delete a user from the system, current incidents no longer display the deleted user as a caller. However, there can be archived incidents that still display the string "John Smith" as the user because the user existed at the time of the archive.

Features

The System Definition Table Rotation module allows you to define a new table rotation, a new table extension or modify an existing one.
Figure 426: Database Rotation

- **Name**: auto-generated from table name
- **Duration**: overall time parameter for function
- **Initialized**: sets function as active (true) or inactive (false)
- **Rotations**: number of tables to be created within Duration
- **Type**: indicates Extension (archiving) or Rotation (deletion) functionality

When you define a new rotation, a schedule is created and new data is subsequently written to one of the tables in the rotation group. You'll notice the that the group includes the original table plus a number of additional tables. Be aware that deleting a rotation will delete the additional tables and all the data, therefore the rotation should not be deleted if the data is needed.

**Implementation**

See individual pages to implement either table rotation or table extension.

**Usage**

Table Rotation can be used for tables that:
- Hold data that is only valuable for a limited period of time
- Are written sequentially by inserting new rows

Table Extension can be used for tables:
- That hold data that's organized by time
- Where data is written and then never updated, or where updates are mostly focused on recent data
- Where data is needed for a period of time (e.g., two years) and can subsequently be discarded or archived off the system
Export data

ServiceNow offers a variety of ways for administrators and users with the itil role to export data.

ServiceNow offers a variety of ways for administrators and users with the itil role to export data:

• **Form export**: Export an individual record from the user interface. Choose PDF or XML format directly from a form.
• **List export**: Export multiple records from the user interface. Choose CSV, Excel, PDF, or XML format directly from a list.
• **Schedule a report**: Automatically export multiple records from a table on a set schedule. Create a scheduled job to regularly export data as a report.
• **Direct URL access**: Export multiple records from a table using the ServiceNow CSV, Excel, PDF, or XML processor. Specify the table form or list you want to export in the URL.
• **Web services/SOAP**: Export multiple records from a table when an external client makes a web services request. Create an external application or process to automate the retrieval of data from an instance via web services or SOAP.

Export and import XML files

To occasionally migrate data from one instance to another, you can export the XML data from one instance and import it to another.

This method ensures that all fields and values are transferred exactly. Migrating data in an XML file saves time for unscheduled data imports since there is no need to build an import set or a transform map.

Exporting and importing data in XML files is commonly used for records created in a development instance that must be migrated with the update sets as part of a migration procedure. Examples of these records include lookup tables, unit test records, and other information required to support production. Typically, this information is only migrated once and the overhead of an import set is not justified.

---

**Note**: Image field data is not preserved when exporting to XML.

---

**Export multiple records as XML data**

Suppose you created a table to look up approvers to support the problem management process.

**Role required**: admin

The table, fields, security, forms, and many other configuration parameters are captured by an update set; however, the data records are not. To promote the lookup records from the development instance to other instances, you must manually export all the records.

1. Sign in to the instance that contains the source data.
2. Navigate to the list you want to export.
3. Optional: Filter the list, if desired.
4. Right-click the list header and select Export > XML.
5. In the export progress dialog box, click **Download** when the export completes.

Depending on browser and settings, a dialog box may prompt you to save the file, or the browser may automatically save the XML file to the downloads folder specified in the browser preferences.

*Export a record as XML data*

How to export a single record as XML data.

**Role required:** admin

It is often useful to export a single record, such as an incident, a user, a configuration item (CI), or a scheduled job, from one instance to another. For example, if a user has issues in a production instance and you want to do in-depth diagnostics on the development instance without impacting the user, you can export the user, CI, or incident record as XML data for later import into the development instance.

1. Sign in to the instance that contains the source data.
2. Navigate to the record you want to export.
3. Right-click the form header and select **Export > XML (This record)**.

Depending on browser and settings, a dialog box may prompt you to save the file, or the browser may automatically save the XML file to the downloads folder specified in the browser preferences.

*Import a single XML file containing one or more records*

After you have successfully exported data from the source instance, you can import the XML file directly to the target instance.

**Role required:** admin

Importing XML does not trigger business rules or update the instance cache.

1. Sign in to the target instance (the instance that should receive the data).
2. In the banner frame, click the **Elevate privileges** icon (🪐) beside the user name.
3. In the **Activate an Elevated Privilege** dialog box, select the **security_admin** check box and click OK.
4. Navigate to any list in the system.
   Any list can be used because the XML file contains the destination table name for the records.
5. Right-click the list header and select **Import XML**.
6. In the import screen, click **Choose File** and select the previously exported XML file.

7. Click **Upload**.

   **Note:** If the data does not import, navigate to **System Definition > Tables & Columns** and verify that the table from which the data was exported also exists in the instance importing the data. If the table does not exist in that instance, you can move it using an update set.

---

**Example XML export**
The URL query produces an XML document similar to this sample.

The following URL query produces an XML document similar to the sample shown:

```plaintext
https://<instance name>.service-now.com/incident_list.do?XML&sysparm_query=priority=1&sysparm_orderby=assigned_to
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xml>
<incident>
<active>true</active>
<activity_due>2012-11-28 20:44:11</activity_due>
<approval>not</approval>
</incident>
</xml>
```
Available export formats

The platform supports certain export formats.

Table 328: Available export formats

<table>
<thead>
<tr>
<th>Export format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>Export table records as a comma-separated value text file. Use this option to export the currently displayed fields in the list or form as a text file. Configure the list or form to add or remove fields from the export. When exported to CSV, dot-walked fields appear using their full field name, such as u_assignment_group.parent.</td>
</tr>
</tbody>
</table>

**Note:** By default, ServiceNow exports all CSV files in Windows-1252 encoding. If you need to export translated data, set the glide.export.csv.charset system property to UTF-8.
<table>
<thead>
<tr>
<th>Export format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excel</td>
<td>Export table records as a Microsoft Excel spreadsheet. Use this option to export the currently displayed fields in the list or form as an Excel spreadsheet. Configure the list or form to add or remove fields from the export.</td>
</tr>
<tr>
<td>XML</td>
<td>Export table records as an XML document. Use this option to export all data from a table or all data for a particular record. The XML file has an XML element for each column in the table.</td>
</tr>
<tr>
<td>PDF</td>
<td>Export table records as a Portable Document Format file. Use this option to export the currently displayed fields in the list or form as a PDF file. Configure the list or form to add or remove fields from the export.</td>
</tr>
</tbody>
</table>

**Note:** The PDF export processor prints data from left-to-right, which can produce rendering errors when displaying right-to-left language data.

**Export format processors**
The ServiceNow platform provides a default upper limit for XML data exports.

ServiceNow provides the following export format processors:

**Table 329: Format processors**

<table>
<thead>
<tr>
<th>Export processor</th>
<th>URL syntax</th>
<th>Export limits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>?CSV</td>
<td>10,000 rows</td>
<td>Exports table records as a comma-separated value text file.</td>
</tr>
<tr>
<td>Excel</td>
<td>?EXCEL</td>
<td>10,000 rows</td>
<td>Exports table records as a Microsoft Excel spreadsheet.</td>
</tr>
<tr>
<td>XML</td>
<td>?XML</td>
<td>10,000 rows</td>
<td>Exports table records as an XML document.</td>
</tr>
<tr>
<td>PDF</td>
<td>?PDF</td>
<td>5,000 rows</td>
<td>Exports table records as a Portable Document Format file.</td>
</tr>
<tr>
<td>Schema</td>
<td>?SCHEMA</td>
<td>N/A</td>
<td>Exports the database schema for the table.</td>
</tr>
<tr>
<td>XSD</td>
<td>?XSD</td>
<td>N/A</td>
<td>Exports the table structure in XSD format.</td>
</tr>
</tbody>
</table>

See Export Limits for information about processor export limits and how to work around them if a table exceeds the export limit.
Note: Export processors return all requested records regardless of whether you use the table.do or table_list.do format to identify the export table.

Export limits

The ServiceNow platform provides a default upper limit for XML data exports. The purpose of the upper limit is to avoid creating performance issues when a table is excessively large. If you need to export more records than the threshold permits, break up the export into separate manageable chunks.

In addition to the format-specific limits, you may need to set com.glide.processors.XMLProcessor.max_record_count to match the upper limit set by the format-specific limit.

Export limit properties

You can set the number of records to return during an export using the URL parameter sysparm_record_count.

However, the system analyzes the following settings to determine whether an export limit should be applied.

1. First, the platform checks the property that defines the format-specific export limit (see table). Each format can have a different limit. Although this property can be set to any value, exceeding the default export limit can impact system performance. You may want to set the property at or below the default limit and have users export large amounts of data in smaller increments.

2. If the format-specific property is not set, the system checks the property for the general export limit (see table). This property can also be set to any value, but exceeding the default export limit can impact system performance.

3. If neither the format-specific export limit nor the general export limit property is set, the system enforces the default export limit (see table).

Note: These properties are not defined by default. You must add the property to assign a value to it.

Table 330: Default export limit

<table>
<thead>
<tr>
<th>Format</th>
<th>Format-specific export limit</th>
<th>General export limit</th>
<th>Default export limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>glide.xml.export.limit</td>
<td>glide.ui.export.limit</td>
<td>10,000</td>
</tr>
<tr>
<td>CSV</td>
<td>glide.csv.export.limit</td>
<td>glide.ui.export.limit</td>
<td>10,000</td>
</tr>
<tr>
<td>EXCEL</td>
<td>glide.excel.export.limit</td>
<td>glide.ui.export.limit</td>
<td>10,000</td>
</tr>
<tr>
<td>PDF</td>
<td>glide.pdf.max_rows</td>
<td>N/A</td>
<td>5,000</td>
</tr>
<tr>
<td>PDF</td>
<td>glide.pdf.max_columns</td>
<td>N/A</td>
<td>25</td>
</tr>
</tbody>
</table>

Although the number of columns can be set higher than 25 in the PDF export, this is not advisable as only 25 header labels fit on a page.

A warning threshold property called glide.ui.export.warn.threshold controls how the records are exported. If a user attempts to export a number of records from a list that exceeds the warning threshold, a dialog box offers the choice of waiting for the export to complete or having the exported records emailed as an
attachment. The warning threshold can be changed in the system property. The email attachment must not exceed the maximum allowed email size or configured email attachment size.

### Table 331: Export limit examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Property</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exporting to CSV</td>
<td>• glide.csv.export.limit = 20,000&lt;br&gt;• glide.ui.export.limit = 10,000&lt;br&gt;• com.glide.processors.XLSXProcessor.max_record_count = 20,000&lt;br&gt;• Default export limit for CSV = 10,000</td>
<td>Table 332: Exporting to CSV&lt;br&gt;<strong>Table 332: Exporting to CSV</strong>&lt;br&gt;<strong>Records to be Exported</strong>&lt;br&gt;15,000&lt;br&gt;30,000&lt;br&gt;<strong>Records Returned</strong>&lt;br&gt;15,000&lt;br&gt;20,000&lt;br&gt;<strong>Note</strong>&lt;br&gt;In the second export, the number of records returned from the database is limited because the number of records specified for export exceeds the value set in the glide.csv.export.limit property.</td>
</tr>
<tr>
<td>Exporting to Excel</td>
<td>• glide.excel.export.limit = no entry&lt;br&gt;• glide.ui.export.limit = no entry&lt;br&gt;• Default export limit for EXCEL = 10,000</td>
<td>Table 333: Exporting to Excel&lt;br&gt;<strong>Table 333: Exporting to Excel</strong>&lt;br&gt;<strong>Records to be Exported</strong>&lt;br&gt;10,000&lt;br&gt;30,000&lt;br&gt;<strong>Records Returned</strong>&lt;br&gt;10,000&lt;br&gt;10,000&lt;br&gt;<strong>Note</strong>&lt;br&gt;In the second export, the number of records returned from the database is limited because the number of records specified for export exceeds the default export limit for Excel, 10,000 records.</td>
</tr>
<tr>
<td>Exporting to PDF</td>
<td>• glide.pdf.max_rows = 3,500&lt;br&gt;• Default and maximum export limit for PDF = 5,000</td>
<td>Table 334: Exporting to PDF&lt;br&gt;<strong>Table 334: Exporting to PDF</strong>&lt;br&gt;<strong>Records to be Exported</strong>&lt;br&gt;2,000&lt;br&gt;10,000&lt;br&gt;<strong>Records Returned</strong>&lt;br&gt;2,000&lt;br&gt;5,000&lt;br&gt;<strong>Note</strong>&lt;br&gt;In the first export, all records are returned because the number of records specified for export does not exceed the glide.pdf.max_rows property. In the second export, the number of records returned is limited because the number of records specified for export exceeds the value in the glide.pdf.max_rows property.</td>
</tr>
</tbody>
</table>

**Excel export threshold**

Excel exports are intended for relatively small exports, fewer than 500,000 cells, while CSV can handle larger exports.
Whenever you export to Excel and the resultant spreadsheet has more than 500,000 cells (by default), the export process stops and you are given the Excel file at that point. In the bottom row, there will be the following message: Export stopped due to excessive size. Use CSV for a complete export:

![Excel export threshold 1](image)

**Figure 427: Excel export threshold 1**

The Excel export cell threshold is customizable using the `glide.excel.max_cells` integer property.

**Note:** Increasing this threshold may cause a memory issue in your instance. The threshold is set at an appropriate level to prevent resource issues.

The export will put the information into the Excel document with 32,000 rows per spreadsheet.

**Export form data**

How to export form data.

Export an individual record *from a form* by right-clicking a form header bar and selecting the export type. Export formats include:

- PDF (Portrait)
- PDF (Landscape)
- XML (This Record)
Note: When exporting PDF data from a form, only the fields that are visible from the current view are exported, with the exception of formatter elements. When exporting XML data, however, all the fields are exported, regardless of the view. You cannot export records to CSV or Excel from a form.

Export list data
You can export a list of records in a variety of formats.

Export a list of records by right-clicking a list header bar and selecting the export format.
**Figure 429: Export from list**

**Note:** To export records in an **embedded list**, export the **record** containing the list.

Export formats include:

- Excel
- CSV
- XML
- PDF (Portrait)
- PDF (Landscape)
- PDF (Detailed Portrait): Exports the list and the associated form for each record.
- PDF (Detailed Landscape): Exports the list and the associated form for each record.

**Note:** You can control how line breaks appear in exported CSV data using the glide.csv.export.line_break system property.

**Note:** To remove unwanted HTML tags from list data, see the blog *Rendering HTML in Exported Lists* by a ServiceNow employee in the ServiceNow Community.

**Determine which list fields are exported**

By default, exporting data from a list exports only the fields that are visible from the current view.

By default, exporting data from a list exports only the fields that are visible from the current **view**. If you want to export fields from another list view, you can switch views from the UI. Alternatively, you can add the sysparm_view parameter to the **URL request**. For example, to export fields visible from the Self Service (ess) view:

```plaintext
https://instance_name.service-now.com/incident.do?CSV=sysparm_view=ess
```
If you are exporting CSV or Excel data and do not specify a view, the export uses the default list view. You can export all fields by setting the `sysparm_default_export_fields` parameter to all. If you are exporting XML data, the export returns all fields unless you specify a particular view. The `sysparm_default_export_fields` parameter has no effect on XML exports.

**Use a URL query to filter a list result**

Use URL queries to programmatically generate filtered lists before exporting them.

URL queries are useful for scripts that generate a list of records and where no user will manually add the filter from the UI. You must be familiar with the table's column names and values to create a query.

1. Specify the instance URL. For example, `https://demo.service-now.com/`.
2. Specify the list URL for the table you want to export. For example, `incident_list.do`.
3. Specify the export format processor for the export. For example, `?XML`.
4. Specify the query as the value of the `sysparm_query` parameter. For example, `?sysparm_query=priority=1`.
5. Optional: Specify the result sort order with the `sysparm_orderby=` parameter. For example, `CSV&sysparm_orderby=assigned_to`.

**Note:** All queries use a column name, an operator, and a value. See Condition Builder for a list of available operators.

For example, to export a list of all priority 1 incidents as an XML file, use the following URL: `https://<instance name>.service-now.com/incident_list.do?XML&sysparm_query=priority=1&sysparm_orderby=assigned_to`

**Exporting currency fields to Excel**

Exporting currency fields to Excel applies Account formatting and can be configured to convert all values to US dollars.

When exporting currency fields to Microsoft Excel, the cells containing currency data use Accounting formatting. This formatting allows you to perform numeric operations on those values as well as view the currency symbol, such as $ or €.

When exporting currency fields the type of currency, such as $ or €, is preserved by default. You can choose to export all currency values in US dollars by setting the property `glide.excel.fixed_currency_usd` to true. The conversion rates for non-USD currencies are stored on the Exchange Rates [fx_rate] table.

**Export data with scheduled reports**

You can schedule regular or one-time exports from list reports.

You can schedule regular or one-time exports from list reports. ServiceNow sends the report to one or more users by email. Export formats include:

- Excel
- CSV
- PDF
- PDF (Landscape)

If you need to share data with another ServiceNow instance or integrate with another application, consider creating a web service or SOAP message instead.

**Export directly from the URL**

You may want to export data from the URL if you need to dynamically export data from a script or web service.
You must be familiar with the ServiceNow table and column names to export data directly from the URL. See URL schema on page 53 for more information about navigating to forms and lists.

To export data directly from the URL, create a URL containing the following parts:

1. Specify the instance URL. For example, https://<instance name>.service-now.com/.
2. Specify the table form or list to export. For example, incident_list.do.
3. Specify the export format processor to use for the export. For example, ?CSV.
4. Optional: Specify a query and sort order with URL parameters. For example, &sysparm_query=sys_id%3E%3Db4aedb520a0a0b1001af10e278657d27.

The final URL should look like one of these sample URLs:

<table>
<thead>
<tr>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>https://&lt;instance name&gt;.service-now.com/incident_list.do?CSV&amp;sysparm_query=sys_id%3E%3Db4aedb520a0a0b1001af10e278657d27</td>
<td>Export a particular incident to a comma-separated value text file.</td>
</tr>
<tr>
<td>https://&lt;instance name&gt;.service-now.com/incident_list.do?CSV&amp;sysparm_orderby=sys_id</td>
<td>Export all incidents to a comma-separated value text file and sort the list by sys_id.</td>
</tr>
</tbody>
</table>
**Note:** ServiceNow enforces *basic authentication* for direct URL access. The data extracted from the URL contains only the fields to which the currently authenticated user has read access.

**URL query parameters**

ServiceNow provides certain URL query parameters.

ServiceNow provides the following URL query parameters:
### Table 336: URL query parameters

<table>
<thead>
<tr>
<th>URL Parameter</th>
<th>URL Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_query</td>
<td>sysparm_query=[column name] [operator][value]</td>
<td>Displays a list of records that match the query. For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>https://&lt;instance name&gt;.service-now.com/incident_list.do?XML&amp;sysparm_query=priority=1</td>
</tr>
<tr>
<td>sysparm_orderby</td>
<td>sysparm_orderby=[column name]</td>
<td>Sorts a list of records by the column name provided. For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>https://&lt;instance name&gt;.service-now.com/incident_list.do?XML&amp;sysparm_query=priority=1&amp;sysparm_orderby=assigned_to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can sort by only one column using sysparm_orderby. To sort by multiple columns, use sysparm_query=ORDERBY[column name]^ORDERBY[column name]. For example: sysparm_query=ORDERBYassigned_to^ORDERBYpriority.</td>
</tr>
</tbody>
</table>

### Call URL export programmatically

You can dynamically export data from a script or web service by calling a URL export from any programming language.

The following procedure includes code samples that demonstrate calling a URL export in C# for a .Net framework call:

1. **Add the following Imports:**

   ```csharp
   using System.Net;
   using System.IO;
   ```

2. **Call the Download method:**

   ```csharp
   static void Main(string[] args)
   {
     // Call to DownloadFile method supplying the URL and location to save CSV file locally
     int read = DownloadFile("https://<instance>.service-now.com/incident_list.do?CSV&sysparm_query=priority=1&sysparm_orderby=assigned_to", "c:\\test\\incident.csv");
   }
   ```

3. **Create a Download method as follows:**

   ```csharp
   public static int DownloadFile(String url, String localFilename)
   ```

   © 2017 ServiceNow. All rights reserved. 1507
{ // Function will return the number of bytes processed // to the caller. Initialize to 0 here.
int bytesProcessed = 0;
// Assign values to these objects here so that they can // be referenced in the finally block
Stream remoteStream = null;
Stream localStream = null;
WebResponse response = null;
// Use a try/catch/finally block as both the WebRequest and
Stream // classes throw exceptions upon error
try
{
    // Create a request for the specified remote file name
    WebRequest request = WebRequest.Create(url);
    // Create the credentials required for Basic Authentication
    // Add the credentials to the request
    request.Credentials = cred;
    if (request != null)
    {
        // Send the request to the server and retrieve the
        // WebResponse object
        response = request.GetResponse();
        if (response != null)
        {
            // Once the WebResponse object has been retrieved,
            // get the stream object associated with the
            response's data
            remoteStream = response.GetResponseStream();
            // Create the local file
            localStream = File.Create(localFilename);
            // Allocate a 1k buffer
            byte[] buffer = new byte[1024];
            int bytesRead;
            // Simple do/while loop to read from stream until
            // no bytes are returned
            do
            {
                // Read data (up to 1k) from the stream
                bytesRead = remoteStream.Read(buffer, 0, buffer.Length);
                // Write the data to the local file
                localStream.Write(buffer, 0, bytesRead);
                // Increment total bytes processed
                bytesProcessed += bytesRead;
            } while (bytesRead > 0);
        }
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
} finally
{
    // Close the response and streams objects here
    // to make sure they're closed even if an exception
    // is thrown at some point
    if (response != null) response.Close();
    if (remoteStream != null) remoteStream.Close();
    if (localStream != null) localStream.Close();
Break up a large export

If the number of records to be exported exceeds the actual export limit, you may want to break the export into smaller increments that do not place a significant performance load on the platform.

1. **Filter the list** to display the records you want to export.
2. Write down the number of records returned.
3. If the record number is higher than the defined threshold, issue a `sysparm` query for the first 10,000 records using the following syntax:
   
   ```
   https://<instance name>.service-now.com/syslog_list.do?
   XML&sysparm_orderby=sys_id&sysparm_record_count=10000
   
   This exports the first 10,000 records in order, sorted by the sys_id number.
   ```

4. Find the next record in order, such as 10,001.
5. Right-click the row and copy the sys_id of the next record you want to export.
6. Access the next series of records with a greater than or equal to query run against the sys_id of record 10,001.

   The following example shows a query that uses a sys_id of b4ae6b520a0a1001af10e278657d27. Use the syntax shown in this query to export the next set of records.

   ```
   https://<instance name>.service-now.com/syslog_list.do?XML&sysparm_query=sys_id%3E%
   %3Db4ae6b520a0a1001af10e278657d27&sysparm_orderby=sys_id&sysparm_record_count=10000
   ```

   **Note:** URL queries use typical percent encoding. In this example, the greater than sign (>) is encoded as %3E and the equal sign (=) is encoded as %3D.

7. Continue issuing this query, using the starting sys_id for the next set of records until you have exported all the necessary records.

Enable export debug logging

When the property `glide.export.debug` is true, the instance logs export processing including database query time and the time taken to write data to the file.

When the property `glide.export.debug` is true, the instance logs export processing including database query time and the time taken to write data to the file. Debug logs are indicated by the text Export API. Prolonged use of this property can affect performance, so it is best to use it while debugging export processing, and then set the property back to false.

```
7/17/14 15:53:48 (500) EB39A310EB022100C46AC2EEF106FED9 Maximum record count for this instance is: 10000, request is for: 0 Cap the Record count to Maximum Record Count
07/17/14 15:53:48 (522) EB39A310EB022100C46AC2EEF106FED9 Export API - ExportProcessor : Processing EXCEL export request, ExportParameters:TableName=incident, Query=active=true, Limit=10000, SortBy=null
07/17/14 15:53:48 (527) EB39A310EB022100C46AC2EEF106FED9 Export API - ExportProcessor : Export using background thread
07/17/14 15:53:48 (528) EB39A310EB022100C46AC2EEF106FED9 #748 / poll_processor.do -- total transaction time: 0:00:03.357, total wait time: 0:00:00.000, session wait: 0:00:00.000, semaphore wait: 0:00:00.000, source: 0:0:0:0:0:0:0:1%0
```
### Import sets

Import Sets allow administrators to import data from various data sources, and then map that data into ServiceNow tables.

<table>
<thead>
<tr>
<th>Explore</th>
<th>Set up</th>
<th>Administer</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Import and export release notes</td>
<td>• Import Sets key concepts on page 1511</td>
<td>• Schedule a data import on page 1573</td>
</tr>
<tr>
<td>• Upgrade to Geneva</td>
<td>• Create a data source on page 1553</td>
<td>• Standard import set tables on page 1544</td>
</tr>
<tr>
<td></td>
<td>• Create a transform map on page 1556</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use</th>
<th>Develop</th>
<th>Integrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Run an import on page 1571</td>
<td>• Developer training</td>
<td>• Web service import sets on page 1582</td>
</tr>
<tr>
<td>• Schedule a data import on page 1573</td>
<td>• Developer documentation</td>
<td>• Importing from another ServiceNow instance on page 1604</td>
</tr>
<tr>
<td>• Easy import on page 1594</td>
<td>• Field map script variables on page 1564</td>
<td></td>
</tr>
<tr>
<td>• Completed import sets on page 1579</td>
<td>• Data import scripting options on page 1575</td>
<td></td>
</tr>
</tbody>
</table>

**Troubleshoot and get help**

- Blog: Troubleshooting missing data in date fields
- Blog: Troubleshooting truncated data
- Blog: Improve Import Set performance with the coalesce index feature
- Blog: Load All Records vs. Test Load 20 Records
- Search the HI knowledge base for known error articles
- Contact ServiceNow Support
Import Sets is a powerful tool used to import data from various data sources, and then map that data into ServiceNow tables.

The Import Sets table acts as a staging area for records imported from a data source.

**Note:** Data should not be imported in extremely large chunks. Creating an extremely large import set can cause delays and system outages.

A transform map determines the relationships between fields displaying in an Import Set table and fields in an existing ServiceNow table, such as the Incidents or Users table.

Importing sets will skip records when the data in the instance matches the data being imported.

**Note:** Import Sets run as user System. Therefore, Import Sets cannot add data to encrypted fields.

**Import sets terminology**
A dictionary of key Import Set terms.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Set table</td>
<td>A table that acts as a staging location for records imported from a data source prior to transforming those records.</td>
</tr>
<tr>
<td>Data source</td>
<td>A record that defines where to get the data to import. A data source may point to a file, a JDBC-compatible database, or an LDAP organizational unit.</td>
</tr>
<tr>
<td>Transformation</td>
<td>The conversion of data from an import set table to another table according to the rules defined in a transform map.</td>
</tr>
<tr>
<td>Transform map</td>
<td>A set of field maps that define the relationships between fields in an import set and fields on a table, such as Incident. During transformation, data is copied from the Import Set table to the destination table based on the transform map. A single import set field may be mapped to multiple fields on other tables.</td>
</tr>
<tr>
<td>Foreign record insert</td>
<td>A foreign record insert occurs when an import makes a change to a table that is not the target table for that import. This happens when updating a reference field on a table. For example when updating a value for the caller on an incident the import is actually updating the sys_user table.</td>
</tr>
</tbody>
</table>
Supported import formats
You can import data from several different file formats.

Table 338: File formats

<table>
<thead>
<tr>
<th>Format</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>CSV files must comply with the proposed CSV standard in RFC4180. In particular, this means that double quotes may not appear inside fields. The first row of data in an imported CSV file becomes the header row and defines the columns for that import set.</td>
</tr>
<tr>
<td>Excel</td>
<td>Excel files must have the XLS extension. Excel files with the XLSX extension will produce an error.</td>
</tr>
<tr>
<td>XML</td>
<td>XML files must have a consistent XPath for each data row.</td>
</tr>
</tbody>
</table>

ServiceNow can import data from the following external data sources.

Table 339: External data sources

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBC</td>
<td>Some network configurations may require a MID Server.</td>
</tr>
<tr>
<td>LDAP</td>
<td>LDAP imports require a valid transform map.</td>
</tr>
</tbody>
</table>

Data sources
Data sources are used to create an import set so that data can be processed, if necessary, prior to being mapped onto a production table.

Table 340: Data Source Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Data is in a recognized file format, accessible locally or remotely through several file retrieval methods.</td>
</tr>
<tr>
<td>JDBC</td>
<td>Data is in a database, accessible using JDBC. ServiceNow supports Oracle, MySQL, Sybase, DB2 Universal, and MS SQL Server drivers.</td>
</tr>
<tr>
<td>LDAP</td>
<td>Data is in an LDAP server, accessible through the LDAP or LDAPS ports, 389 and 636 respectively.</td>
</tr>
</tbody>
</table>

File type data sources
You can import a file from a local source, a remote network server, or another instance by providing a path and authentication information.
For file type import sets, you can select from a list of file retrieval methods, including FTP, FTPS, SCP, and HTTPS.

File retrieval methods

The following file retrieval methods are available to copy the file from where it resides to ServiceNow to be loaded into an import set.

After defining the files that are compatible for importing, define how the files can be imported.

**Note:** For the import to succeed, your FTP server and client must be set up for the same authentication mechanism that you select here. Refer to the following article for an explanation of the supported protocols [http://en.wikipedia.org/wiki/FTPS](http://en.wikipedia.org/wiki/FTPS).

<table>
<thead>
<tr>
<th>Table 341: File Retrieval Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attachment</strong></td>
</tr>
<tr>
<td><strong>File</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>SFTP</strong></td>
</tr>
<tr>
<td><strong>FTP</strong></td>
</tr>
<tr>
<td><strong>FTPS (Auth SSL)</strong></td>
</tr>
<tr>
<td><strong>FTPS (Auth TLS)</strong></td>
</tr>
</tbody>
</table>
FTP data source extended properties

Certain properties are available for customizing your FTP data source.

To enter these properties, you must configure the Data Source form and add the Properties field. When specifying more than one property, use a comma to separate each property.

### Table 342: Available FTP Properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Default value</th>
<th>Example custom value</th>
<th>Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection.passive</td>
<td>Flag that indicates the behavior of the FTP connection. true = passive connection false = active connection</td>
<td>true</td>
<td>connection.passive=false</td>
<td>FTPS</td>
</tr>
</tbody>
</table>
### File data source examples

These examples describe how to import various file types as data sources.

#### XLS data source file

XLS refers to the Microsoft Excel file format. This is the default format for spreadsheets created in Excel for versions 2003 and older. The .xlsx file format that is the default format for Excel 2007 is not currently supported. If importing a list from Excel 2007, it is necessary to save the file as .xls.

**Note:** All .xls files must use the 1900 date system as opposed to the 1904 date system as the latter will cause your dates to be imported with a year that is four years earlier than what your spreadsheet displays. Refer to MS Support for additional information about date systems.
XML data source file

XML data source definitions have an additional field to specify the "XPath for each row". This is an XPath expression to select the elements whose children will be converted to rows in the import set table. Each selected element will cause a row to be created in the import set table. The children of the selected element will be converted into columns in the row. For example, to import the asset information from the sample XML file below, the XPath expression should be specified as "/export/asset". This expression matches the 3 <asset> elements in the file, so 3 rows will be created. Although one could also specify "//asset" this expression is much less efficient for large files. It is strongly recommended that you use the absolute location path form for XPath expressions when using the XML loader. In other words, avoid expressions containing "//" unless strictly necessary.

Be careful not to specify an insufficiently restrictive XPath expression with a large XML file. For example, the expression "/export" would be wrong for the sample file below, since it matches the document (root) element. In general, you should never specify an XPath expression which matches the root element unless you want everything in the document to be made into a single row. Specifying an insufficiently restrictive XPath expression when attempting to load large XML files could result in lengthy processing times and incorrect output. For this reason, it is best to test and debug XML data source specifications with small XML files containing only a few rows worth of data. Once the specification has been tested, you can run it again with the full file.

Notice that the <userInfo> elements contained within <asset> contain child elements. This will cause a column called "userInfo" to be created containing XML for the userInfo element. If Expand child nodes is checked, individual columns will also be created for userInfo/lastName and userInfo/firstName. If Expand child nodes is not checked, only the userInfo XML column will be created.
Note that when loading data from an XML file, ServiceNow samples the first 10 records to determine what fields are required to hold the data. If none of the first 10 records specify a value for a field, ServiceNow does not add that field to the table that holds the imported data. Ensure that at least one of the first 10 records specifies a value for any fields being imported.

```xml
<?xml version= "1.0" encoding= "utf-8" ?>
<export>
  <asset><assetTag>AT-01939</assetTag><type>Desktop</type><os>Windows 7 Professional</os><lastlogondate>12-07-2010 12:31:24</lastlogondate><userInfo><lastName>Loo</lastName><firstName>David</firstName></userInfo><isenabled>true</isenabled></asset><asset><assetTag>AT-53480</assetTag><type>Desktop</type><os>Windows 7 Professional</os><lastlogondate>09-07-2010 13:25:53</lastlogondate><userInfo><lastName>Merritt</lastName><firstName>Norris</firstName></userInfo><isenabled>true</isenabled></asset><asset><assetTag>AT-55782</assetTag><type>Desktop</type><os>Unknown</os><lastlogondate>01-01-1900 00:00:00</lastlogondate><userInfo><lastName>Currie</lastName><firstName>Mike</firstName></userInfo><isenabled>true</isenabled></asset>
</export>
```

**CSV data source file**

Character-separated value (CSV) files are used as a cross-compatible file format for transferring files across platforms. A CSV file is a text file that defines a grid, where columns are defined by commas and rows are defined by line breaks. To define precise spacing for importing strings you can optionally wrap text in quotes.

```
"user_name","name","email","sys_created_on","active"
"jared.laethem","Jared Laethem","jared.laethem@yourcompany.com","2008-02-24 22:21:32","true"
"jerrod.bennett","Jerrod Bennett","jerrod.bennett@yourcompany.com","2007-08-12 12:12:18","true"
"eric.schroeder","Eric Schroeder","eric.schroeder@yourcompany.com","2007-07-03 11:50:20","true"
"rob.woodbyrne","Rob Woodbyrne","rob.woodbyrne@yourcompany.com","2007-07-03 11:49:57","true"
"admin","System Administrator","admin@yourcompany.com","2007-07-03 11:48:47","true"
"christen.mitchell","Christen Mitchell","christen.mitchell@yourcompany.com","2007-05-16 15:26:42","true"
"rob.phillips","Rob Phillips","rob.phillips@yourcompany.com","2007-01-22 11:25:34","true"
"davin.czukowski","Davin Czukowski","davin.czukowski@yourcompany.com","2006-07-11 14:01:26","true"
"luke.wilson","Luke Wilson","luke.wilson@yourcompany.com","2006-02-07 15:29:48","true"
"bow.ruggeri","Bow Ruggeri","bow.ruggeri@yourcompany.com","2005-07-07 11:39:58","true"
"don.goodliffe","Don Goodliffe","don.goodliffe@yourcompany.com","2005-05-02 12:28:40","true"
"david.loo","David Loo","david.loo@yourcompany.com","2005-02-22 16:00:00","true"
"guest","Guest","guest@yourcompany.com","2004-05-01 17:00:00","true"
"fred.luddy","Fred Luddy","fred.luddy@yourcompany.com","2004-05-01 17:00:00","true"
```

When using CSV files, you can specify the encoding charset using the **Properties** field on the Data Source form. You may need to **configure** the Data Source form to see this field. For example, to use utf-8 encoding, enter **charset=utf-8**.

Processing custom CSV files
You can process CSV files that are delimited by a character other than commas.

This is an advanced step to create a CSV import. Normally, you would upload the data and import it directly using System Import Sets, which will create this CSV data source for you automatically.

1. Create the data source record.
2. Attach the CSV file to the data source.
3. Customize the Data Source form and add the **CSV Delimiter** field.
4. Enter the character you want to use as the CSV file delimiter, such as the pipe symbol (|).
5. Test load the data source.

---

### JDBC type data source

A JDBC data source retrieves its data via a JDBC driver, usually type 4 network.

The JDBC connection is available either directly from the ServiceNow instance (a VPN setup required), or via a dedicated MID Server installed inside your firewall that can access the database port.

---

**Note:** Any JDBC call from the MID Server is never encrypted. Limit the rights available to the MID Server user whenever making JDBC calls from a MID server.

**Caution:** The system loads JDBC drivers from the `lib` directory before it looks for the driver in the `extlib` directory. If you synched an updated version of a JDBC library into `extlib`, and the `lib` directory contains an outdated JDBC library, the instance loads the old library from the `lib` directory.

---

### Supported database connections

JDBC connections are supported for the following default databases:
### Table 343: Supported JDBC connections

<table>
<thead>
<tr>
<th>Database</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>3306</td>
</tr>
<tr>
<td>MS SQL Server</td>
<td>1433</td>
</tr>
<tr>
<td>Oracle</td>
<td>1521</td>
</tr>
<tr>
<td>Sybase</td>
<td>5000</td>
</tr>
<tr>
<td>DB2 Universal</td>
<td>&gt;50000</td>
</tr>
</tbody>
</table>

**Note:** Oracle DATE fields are loaded as ServiceNow datetime fields.
JDBC data source SQL statement requirements

JDBC queries that run SQL statements must specify a column name.

For example, this query specifies the column name `ServerID`.

```sql
SELECT DISTINCT 'server_name:' + CONVERT ( VARCHAR , lg .ResourceID ) AS 'ServerID'
,LastHWScan
,Account0
,Category0
Fromv_GS_WORKSTATION_STATUSsInner  JOIN
v_GS_LocalGroupMembers0lg
```

© 2017 ServiceNow. All rights reserved.
SQL server integrated authentication for Windows
A JDBC data source can use the ID of for the Windows MID Server service user to authenticate with SQL Server.

The **Use integrated authentication** check box on the Data Source form determines if the JDBC data source uses the MID Server service user credentials. For this field to appear on the Data Source form, the integration must meet these criteria:

- The MID Server must be running on a Windows computer with SQL Server.
- The MID Server service must use the same credentials that SQL Server requires.
- The data source **Type** is JDBC.
- The data source **Use MID Server** field must not be empty.
- The data source **Format** is SQLServer.

**Important:** If integrated authentication is chosen, and you are integrating with Microsoft System Center Configuration Manager, the account running the MID Server service must have read rights on the SCCM database.
**Figure 431: Data Source form**

JDBC data source connection properties

You can control how ServiceNow communicates with a JDBC data source using properties.
### Table 344: JDBC Connection Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.jdbcprobeloader.retry</td>
<td>The number of times a JDBC probe loader attempts to process data returning from a JDBC data source. Sleeps on the value defined in &quot;glide.jdbcprobeloader.retry.millis&quot; between retries.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 60</td>
</tr>
<tr>
<td></td>
<td>• Location: Add the property</td>
</tr>
<tr>
<td>glide.jdbcprobeloader.retry.millis</td>
<td>How many milliseconds a JDBC probe loader waits in between retry attempts to process data from a JDBC data source.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5000</td>
</tr>
<tr>
<td></td>
<td>• Location: Add the property</td>
</tr>
</tbody>
</table>

**JDBC drivers for unsupported sources**

Use the following procedures to extend the data source to use a driver for a database that is not provided in the base system.

**Install a driver on a MID Server**

You can install a new JDBC driver JAR file to a MID server to access database formats that are not supported by default.

Role required: agent_admin

1. Navigate to **MID Server > JAR Files**.
2. Click **New**.
3. Complete the following fields:
   - **Name**: A unique and descriptive name for identifying the file in the instance.
   - **Version**: A version number for the file, if one is available.
   - **Source**: Location of the JAR file for reference purposes. Source information is not used by the system.
   - **Description**: Short description of the JAR file and its purpose in the instance.
4. Click the paper clip icon in the banner and attach the JAR file to the record.
5. Click **Submit**.
6. Restart the MID Server service.
   
The platform makes the JAR file available to any MID Server configured to communicate with the instance.

Add a new JDBC format choice

Extend the available JDBC driver options by creating a new choice list entry to specify the JDBC driver Java package name.

Role required: import_admin

Add a new choice list entry for the new database to the **Format** field in the sys_data_source table.

1. Navigate to **System Import Sets > Administration > Data Sources**.
2. Click **New**.
3. In the Data Sources form, right-click the **Format** field label, and select **Show Choice List** from the pop-up menu.

4. Click **New** in the list of choices.
5. Provide the following values to create the new database choice. Look at the existing drivers for examples.
- **Table**: sys_data_source
- **Label**: Database name that appears as an option in the **Format** choice list, for example: TeraData.
- **Value**: Package name and class of the driver. For example, the value for TeraData is `com.ncr.teradata.TeraDriver`.
- **Dependent value**: JDBC

6. Click **Submit**.
   The new data source now appears in the list of available JDBC formats.

Add Sybase or DB2 JDBC format choices
- Extend the available JDBC driver options by activating the Sybase or DB2 choices.

Roles required: admin

**Important**: To use Sybase or DB2 JDBC drivers, you must manually install the driver JAR file on a MID server. Refer to [KB0551236](#) for more information.

Activate the Sybase or DB2 JDBC drivers to connect with those database formats through a MID server. This procedure assumes you are using the following Sybase or DB2 drivers:
Table 345: Drivers

<table>
<thead>
<tr>
<th>Format</th>
<th>Driver</th>
<th>Driver class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybase</td>
<td>jconnect-jdbc</td>
<td>com.sybase.jdbc3.jdbc.SybDriver</td>
</tr>
<tr>
<td>DB2</td>
<td>db2jcc</td>
<td>com.ibm.db2.jcc.DB2Driver</td>
</tr>
</tbody>
</table>

If you are using a different implementation of the Sybase or DB2 driver, you must modify the choice Value to use the driver class from your driver implementation.

1. Navigate to System Import Sets > Administration > Data Sources.
2. Click New.
3. In the Type field, select JDBC.
4. Right-click the Format field label, and select Configure Choices from the pop-up menu.
5. Move the Sybase or DB2 Universal choices from the Available column to the Selected column.
6. Click Save.

The new data source now appears in the list of available JDBC formats.

Create a new data source

Add a new JDBC data source by defining a data source record.

Role required: import_admin

1. Navigate to System Import Sets > Administration > Data Sources and click New.
2. In the Data Source form, select JDBC from the list in the Type field.
3. Select the new data source from the choice list in the Format field.
4. Configure the form and add the **Connection URL** field.

5. Provide the connection URL to the JDBC data source, and then click **Submit**.

No other connection information is required.

![Data Source](image)

**LDAP type data source**

An LDAP data source is automatically created when you configure your instance to integrate with LDAP.
Transform maps
A transform map is a set of field maps that determine the relationships between fields in an import set and fields in an existing ServiceNow table, such as Incident [incident] or User [sys_user].
After creating a transform map, you can reuse it to map data from another import set to the same table.

The **Transform Maps** module enables an administrator to define destinations for imported data on any tables. Transform mapping can be as simple as a drag and drop operation to specify linking between source fields on an import set table and destination fields on any table. Use transform mapping to map source and destination fields dynamically.

**Using multiple transform maps**
- Multiple transform maps can be applied to a single data source.
- One import set row is created per transform map. This behavior can cause a large number of temporary records to be generated.

**Run multiple transforms off a single import set**
- Users can select multiple transform maps during data import.
- The selected transform maps will be executed on the same import set in the order specified.
Transform map scripts
Transform Map scripts allow you to customize import operations using a robust programming interface to introduce advanced logic.

A transform map script executes as events occur while an import set is being transformed onto a ServiceNow table. Transform Map scripting is fully integrated into the ServiceNow scripting environment. There are two types of Transform Map scripts:

• Explicit Transform Map scripts, which explicitly define mapping relationships
• Transformation Event scripts, which modify the processing of events at different stages of a transformation

Transformation script variables
Multiple variables can be used to define explicit mapping relationships in a transform map script.

**Variable name: source**
Type: GlideRecord object
Description: Contains the import source record currently being transformed. Specify a specific field from the source record as an object property.

Example:
```
var x = source.incident_state;
```

**Variable name: target**
Type: GlideRecord object
**Variable name: map**

**Type:** GlideRecord object

**Description:** Contains the transformation map record currently being used for the transformation process. Specify a specific field from the transform map record with one of these properties.

- name
- sys_id
- source_table
- target_table
- order

**Example:**

```
var x = map.order;
```

---

**Variable name: log**

**Type:** Function

**Description:** Log information about the current import process. Each log level has its own method.

**Example:**

```
log.info("This is an information message");
log.warn("This is a warning message");
log.error("This is an error message");
```

---

**Variable name: action**

**Type:** Function

**Description:** Specify the transformation action occurring on the target record. This value can be either "insert" or "update".

**Example:**

```
if(action =="insert"){
    ignore = true;
}
```

---

**Variable name: ignore**

**Type:** Boolean

**Description:** When set to true, skips or aborts the current import action. In onStart scripts, this variable aborts the entire transformation process. In onBefore scripts, this variable only skips the current row being transformed.
Example:

```java
if(source.u_user_name.nil()){
    ignore = true;
}
```

Variable name: error

**Type:** Boolean

**Description:** When set to true, aborts the current import action and logs an error message in the Import Set Log.

Example:

```java
if(source.name=="no_transfrom"){
    error = true;
}
```

Variable name: error_message

**Type:** String (output message)

**Description:** When an error occurs, adds the specified error message to SOAP response.

Example:

```java
if(source.name=="no_transfrom"){
    error = true;
    error_message = "Source is not intended for transformation";
}
```

Variable name: status_message

**Type:** String (output message)

**Description:** Adds the specified status message to SOAP response.

Example:

```java
if(action =="insert"){
    status_message = "Inserting record";
}
```

Map with explicit transform map scripts

Explicitly define mapping relationships in the Transform Map record itself.

An explicit Transform Map script explicitly defines mapping relationships in the Transform Map record itself. It runs after the source field values have been copied over to the target record, and before they are written to the database.

CMDB software import facilitated by explicit mapping script

An example demonstrating how explicit map scripting can be used to facilitate an import of software instances to the CMDB where it is necessary to have the count in the software packages table reflect the number of instances.

In the CMDB there exist tables for software packages and software instances. A software package refers to a one individual software title, such as Mozilla Firefox. A software instance refers to an individual instance of a software title such as Mozilla Firefox on Jared_T60_Laptop.

In more technical terms a software instance is a many-to-many relationship with a software package and a configuration item. This script imports software instances. In doing so it checks to see if a software
package for the instance exists. If the software package doesn’t exist then it is created and the sys_id for the newly created package is associated with the instance via the instance record’s Software field.

```javascript
//First we will initialize some temporary variables, referencing
//values from the Import Set table source that will be used to
//reference software package records and create them if necessary.

var name = source.u_name;
var version = source.u_version;
var sid = "";

//Next we will perform a glide query on the software package table (Note: The target table
//for the import is software instances)
var sgr = new GlideRecord("cmdb_ci_spkg");

//Here we are building our query to search for software packages where the
//name and version
//of the package matches the name and version of the instance being
//imported
sgr.addQuery("name", name);
sgr.addQuery("version", version);
sgr.query();

//Now if a software package with the correct name and version are found then
//we
//record the sys_id of the package record
//otherwise we create the package and then record the sys_id
if (sgr.next()) {
    sid = sgr.sys_id; } else { // create it
    sgr.initialize();
    sgr.newRecord();
    sgr.name = name;
    sgr.version = version;
    sid = sgr.insert();

    //Here we make an entry in System Import Set Log saying that we had to
    //create a software package
    log.info("Created new spkg: " + name + " - " + version);
}

//Finally we set the reference field on our software instance record to the
//sys_id we have
//recorded for the software package. In doing so we are also relating the
//software
//package with the instance and so the count, which keeps track of
//the number of instances associated with a package, will automatically be
//incremented.
    target.software = sid;
```

Populating child tables by setting class names in an import to the CMDB

An example demonstrating how to import servers into the configuration management database.

The destination table for the import is the cmdb_ci_servers table.

```javascript
var operating_system = source.u_operating_system.toString();

//This if statement uses JavaScript regular expressions to search the
//operating system
if (operating_system.match(/linux/i) != null) {
    target.sys_class_name = "cmdb_ci_linux_server";
}
if (operating_system.match(/win/i) != null) {
```
User import data sanitation

An example script demonstrating how to sanitize user data before import.

```
var name = source.u_name.toString();

// Use native JavaScript function split to create an array for each word in
// the name "splitting"
// it anywhere that there is a space
var split_names = name.split(" ");

// Find the number of of names (i.e., first and last name only, or first
// middle and last name, etc.)
var num_names = split_names.length;

// If there is only one name then map it to the last name field in the user
// table
if(num_names ==1){
    target.last_name = split_names[0];}

// If there are two names then map the first one to first name and the last
// one to last name
if(num_names ==2){
    target.first_name = split_names[0];
    target.last_name = split_names[1];
}

// If there are more than 3 names then all middle names get combined into one
// middle name
// this is done by shifting off the first name (array element 0 )
// and mapping to first name and popping off the last
// name and returning it to the last name field
if(num_names >=3){
    target.first_name = split_names.shift();
    target.last_name = split_names.pop();
    target.middle_name = split_names.join(" ");}
```

Map with transformation event scripts

Transformation events occur during the process of transforming an import set table onto a table.

These events modify the transformation behavior from any type of mapping specification. Transformation Event scripts modify the processing of the events at various stages of the transformation.

For example, the processing of a mapping operation defined using the Mapping Assist Utility can be manipulated using the event scripts. There are a number of import set JavaScript objects that are accessible during these events. These objects represent tables or portions of tables. It is important to note that what these objects refer to varies depending on the context of the event in which they are referenced.

The following is a table of all of the transform events and their contextual variables:
Table 346: transform events and their contextual variables

<table>
<thead>
<tr>
<th>Event name</th>
<th>Event Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>onStart</td>
<td>When: The onStart event script is processed at the start of an import run, before any data rows are read.</td>
</tr>
</tbody>
</table>

Table 347: Import Set JS objects available to be referenced and their context in the onStart event

<table>
<thead>
<tr>
<th>Import Set JS object</th>
<th>Type</th>
<th>Context in the onStart import set event</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The first row of the source table, there is no data yet since the row has not been read.</td>
</tr>
<tr>
<td>import_set</td>
<td>GlideRecord</td>
<td>The import set that is currently being transformed.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransformMap</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run. For example, log.info(...), log.warn(...), log.error(...).</td>
</tr>
<tr>
<td>ignore</td>
<td>Boolean</td>
<td>When set to true, the entire transformation will be stopped and no further processing will occur.</td>
</tr>
<tr>
<td>error</td>
<td>Boolean</td>
<td>When set to true, has the same effect as the ignore flag of stopping with an error.</td>
</tr>
<tr>
<td>Event name</td>
<td>Event Parameters</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>onComplete</td>
<td>When: The onComplete event script is processed at the end of an import run, after all data rows are read and transformed.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 348: Import Set JS objects available to be referenced and their context in the onComplete event

<table>
<thead>
<tr>
<th>Import Set JS object</th>
<th>Type</th>
<th>Context in the onComplete import set event</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The last row of the source import set table.</td>
</tr>
<tr>
<td>target</td>
<td>GlideRecord</td>
<td>The last row of target table.</td>
</tr>
<tr>
<td>import_set</td>
<td>GlideRecord</td>
<td>The import set that is currently being transformed.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransformMap</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run. For example, log.info(...), log.warn(...), log.error(...).</td>
</tr>
<tr>
<td>error</td>
<td>Boolean</td>
<td>When set to true, will mark the current import set status to error after it completes.</td>
</tr>
</tbody>
</table>

**Example:**

```javascript
// Create a myimport_completed event that can be reacted by an email notification or script action. (there is already an import.finished event that the system will create at the end of an import)
var e = new GlideEvent("myimport_completed", import_set.sys_id, map.sys_id, "");
e.insert();
```

---

**Note:** GlideEvent replaces Packages.com.glide.policy.Event
Event name | Event Parameters
---|---
onBefore | When: The onBefore event script is processed at the start of a row transformation, before the source row is transformed into the target row.

### Table 349: Import Set JS objects available to be referenced and their context in the onBefore event

<table>
<thead>
<tr>
<th>Import Set JS object</th>
<th>Type</th>
<th>Context in the onBefore import set event</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The row of the source table that is currently being processed.</td>
</tr>
<tr>
<td>target</td>
<td>GlideRecord</td>
<td>The row of the target table that is currently being processed.</td>
</tr>
<tr>
<td>import_set</td>
<td>GlideRecord</td>
<td>The import set that is currently being transformed.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransformMap</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run. For example, log.info(...), log.warn(...), log.error(...).</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>Action returns a value of either &quot;insert&quot; or &quot;update&quot; indicating whether the current target row is about to be created or updated.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Event name</th>
<th>Event Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>onAfter</td>
<td>When: The onAfter event script is processed at the end of a row transformation, after the source row has been transformed into the target row and saved.</td>
</tr>
</tbody>
</table>

**Table 350: Import Set JS objects available to be referenced and their context in the onAfter event**

<table>
<thead>
<tr>
<th>Import Set JS object</th>
<th>Type</th>
<th>Context in the onAfter import set event</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The row of the source table that is currently being processed.</td>
</tr>
<tr>
<td>target</td>
<td>GlideRecord</td>
<td>The row of the target table that is currently being processed.</td>
</tr>
<tr>
<td>import_set</td>
<td>GlideRecord</td>
<td>The import set that is currently being transformed.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransform</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run. For example, log.info(...), log.warn(...), log.error(...).</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>Action returns a value of either &quot;insert&quot; or &quot;update&quot; indicating whether the current target row was created or updated.</td>
</tr>
<tr>
<td>Event name</td>
<td>Event Parameters</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>onForeignInsert</td>
<td>When: The onForeignInsert event script is processed at the start of the creation of a related, referenced record, before the record is created.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 351: Import Set JS objects available to be referenced and their context in the onForeignInsert event**

<table>
<thead>
<tr>
<th>Import Set JS object</th>
<th>Type</th>
<th>Context in the onForeignInsert import set event</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The row of the source table that is currently being processed.</td>
</tr>
<tr>
<td>target</td>
<td>GlideRecord</td>
<td>The row of the target table that is currently being processed.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransformMap</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run. For example, log.info(...), log.warn(...), log.error(...).</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>Action returns a value of &quot;insert&quot; or &quot;update&quot; indicating whether the current target row is about to be created or updated.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>Evaluates to the field name of the target record for which a</td>
</tr>
<tr>
<td>Event name</td>
<td>Event Parameters</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>onChoiceCreate</td>
<td></td>
<td>When: The onChoiceCreate event script is processed at the start of a choice value creation, before the new choice value is created.</td>
</tr>
</tbody>
</table>

Table 352: Import Set JS objects available to be referenced and their context in the onChoiceCreate event

<table>
<thead>
<tr>
<th>Import Set JS object</th>
<th>Type</th>
<th>Context in the onChoiceCreate import set event</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The row of the source table that is currently being processed.</td>
</tr>
<tr>
<td>target</td>
<td>GlideRecord</td>
<td>The row of the target table that is currently being processed.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransformMap</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, log.info(...), log.warn(...), log.error(...).</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>Action returns a value of either &quot;insert&quot; or &quot;update&quot; indicating whether the current target row is about to be created or updated.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>Evaluates to the field name of the target record for which a choice value is about to be created.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>Evaluates to the display value from the source record for which a choice is about to be created.</td>
</tr>
</tbody>
</table>
### Event name | Event Parameters
---|---
**onReject** | When: The onReject event script is processed during the occurrence of a foreign record or choice creation, and the foreign record or choice is rejected, the entire transformation row is not saved.

#### Table 353: Import Set JS objects available to be referenced and their context in the onError event

<table>
<thead>
<tr>
<th>Import Set JS object</th>
<th>Type</th>
<th>Context in the onError import set event</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The row of the source table that is currently being processed.</td>
</tr>
<tr>
<td>target</td>
<td>GlideRecord</td>
<td>The row of the target table that is currently being processed.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransformMap</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>Action returns a value of either &quot;insert&quot; or &quot;update&quot; indicating whether the current target row is about to be created or updated.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run. For example, log.info(...), log.warn(...), log.error(...).</td>
</tr>
</tbody>
</table>

Example:

```javascript
//Create an event
var e = new GlideEvent("myimport_onReject", action, "", ");
e.insert();
```
Updating records using coalesce

The coalesce option allows you to update existing target table records when transforming import data.

The coalesce option on a field map allows you to specify if the selected **Target field** should be used to coalesce on when import set records are transformed. If the field map **Coalesce** checkbox is selected, when the import set row is transformed the instance checks for an existing record in the target table that has the same value in the **Target field** as the import set row **Source field**.

If an existing record with a matching value in the target table is found, that record is updated. If no matching record is found, then a new record is created in the target table.

**Note:** Choose fields in the target table to coalesce on only if those fields will have unique values. If more than one record in the target table matches the specified coalesce options, only the first matching record in the target table is updated.

Coalesce options

There are several possible configurations you can use to coalesce data in import sets.

**No coalesce**

If no coalesce is defined, all imported rows are treated as new records. No existing records are updated.

**Single-field coalesce**

You can coalesce on a single field to update an existing record.

If a target table record exists with the same value in the coalesce field as the staging table record, the target table record is updated using the import set record values.

**Multiple-field coalesce**

You can coalesce on multiple fields to update an existing record.

If a target table record exists with the same values in all coalesce fields as the staging table record, the target table record is updated using the staging table record values. All coalesce field values between the target and staging tables must match to coalesce with multiple fields.

**Conditional coalesce**

You can use a script to determine if a staging table row should coalesce to a target record.

Most conditional coalesce scripts are defined in the **Source script** field of a field map for the sys_id field. To update a target record using the staging table record values, the script must return the sys_id of the target table record.

**Example conditional coalesce scripts**

Review examples of conditional coalesce scripts.
Updates only

To only update records where a match is found, and skip records where a match is not found, specify a coalesce field and add the following script as an `onBefore` script to the transform map.

```javascript
if (action == 'insert')
    ignore = true;
```

Conditional coalesce using dot-walking

You can use dot-walked fields in a conditional coalesce script, such as to match the email address of a user when importing incident data.

In this example, this script is defined in the **Source script** of a field map for the Incident target table `sys_id` field.

```javascript
var gr = new GlideRecord('incident');
gr.addQuery('caller_id.email', source.u_email); //check if the incident caller's email matches the import row email value
gr.query();
if (gr.next())
    answer = gr.sys_id; //if a match exists, return the sys_id of the matching Incident record
else
    answer = -1;
```

Conditional coalesce with an OR condition

You can use a conditional coalesce script to match source and target records based on multiple field values. Unlike multiple-field coalesce where all coalesce fields must match, you can specify a script to only require one of the fields to match.

In this example, this script is defined in the **Source script** of a field map for the User target table `sys_id` field.

```javascript
var gr = new GlideRecord('sys_user');
var qc = gr.addQuery('email', source.u_email); //first check if the user's email matches
qc.addOrCondition('user_name', source.u_name); //alternatively, check if the username matches
gr.query();
if (gr.next())
    answer = gr.sys_id; //if a match exists, return the sys_id of the matching User record
else
    answer = -1;
```
Case sensitive coalesce field values

Coalescence is the method by which import sets update existing records in a destination production table, rather than inserting new records.

This means that when importing the data onto the targeted production table, the import set application attempts to match source values to currently existing target values on a production table for updating. Otherwise, a new record is created.

A field called **Coalesce case sensitive** in the Field Map form (System Import Sets > Transform Maps > Field Map), enables you to coalesce field values by case sensitivity.

**Case sensitivity in coalescence**

By default, field values marked as Coalesce are used in a case insensitive lookup for existing records. Case insensitive records update existing records only, and do not cause new records to be created. If this check box is selected, the ServiceNow system attempts to match coalesce field values in import sets by case. If the case of a field in a record in the import set does not match the case in a value in an existing record, a new record is created.

**Figure 434: Case sensitivity in coalescence**

Coalesce on empty fields

You can control if an import set will coalesce on fields with no value.

The field map **Coalesce empty fields** field allows you to coalesce on fields with no value.

By default, fields marked as Coalesce in the field map cause the import set to check for a target record with a field value that matches the value from the import set staging record. When **Coalesce empty fields** is selected for that field map, an empty value in the target record and import set staging record counts as a match for the purpose of coalescing.

For example, the User transform map coalesces on the email field. With the **Coalesce empty fields** option selected, a source record containing an empty email address coalesces to a target record containing an empty email address.

**Standard import set tables**

Several Import Set tables are available by default.

**Notification**

A standard object for describing an external interface for a notification in the system, eg. alarms and alerts from monitoring systems. The default transform map for this object will create or update an incident record. The incoming notifications are coalesced into incidents based on the UUID field.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuid</td>
<td>The universally unique identification number or string that uniquely identifies this notification. It is marked as the coalescing value in the default transform map for the corresponding Incident and is mapped to the correlation_id field of Incident.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>corrective_message</td>
<td>A free form string value that indicates the corrective or followup steps to be taken to address the issue identified in this notification. This field is not mapped by default.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>duration</td>
<td>A string value representing the time value duration affecting the issue reported in this notification. Out of box, the duration field is not mapped. The format of the time is up to the calling program and must be mapped accordingly in the default map to be used.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>expires_on</td>
<td>A string value representing the datetime value that the issue reported in this notification will expire. Out of box, the expires_on field is not mapped. The format of the time is up to the calling program and must be mapped accordingly in the default map to be used.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>message</td>
<td>A string value describing the nature of the issue related to this notification. It should be a concise description and is mapped to the short_description field of the Incident.</td>
<td>Character (80)</td>
</tr>
<tr>
<td>comments</td>
<td>A string value containing additional comments related to this notification. The value is mapped to the comments field of the Incident.</td>
<td>Character (4000)</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Data type</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>category</td>
<td>A string value categorising the nature of this notification. The value is mapped to the category field of the Incident, and therefore should be one of its valid values. If an existing value does not exist, the default behavior is to create a new category.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>assignment_group</td>
<td>A string value of the assignment group for this notification. The value is mapped to the assignment_group field of the Incident, and therefore should be one of its valid values. If an existing value does not exist, the default behavior is to create a new assignment group and set it for the resulting incident.</td>
<td>Character (40)</td>
</tr>
</tbody>
</table>
| severity     | A string representation of a numeric value that indicates the severity of the issue being reported in this notification. This field is mapped to the severity field on Incident. The out of box numeric values and their meanings are:  
  • 1 - High  
  • 2 - Medium  
  • 3 - Low | Character (40)     |
| state        | A string that indicates the state of the issue being reported in this notification. This field is mapped to the incident_state field on Incident. The out of box values are:  
  • New  
  • Active  
  • Resolved  
  • Closed | Character (40)     |
<p>| source       | A string value to indicate the source of the issue or the configuration item (by unique identifier eg IP address, host name etc) related to the issue in this notification. It is mapped to the cmdb_ci field of Incident. | Character (40)     |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>A string value representing the datetime value that marks the beginning of the issue reported in this notification. Out of box, the timestamp field is not mapped. The format of the time is up to the calling program and must be mapped accordingly in the default map to be used.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>type</td>
<td>A string value categorizing the type of issue related to this notification. Out of box, this field is not mapped to any field on Incident. Integrations using this Notification message may use this field to identify its source and trigger additional scripts.</td>
<td>Character (40)</td>
</tr>
</tbody>
</table>

**Computer**

A standard object for describing an external interface for a computer in the system. The default transform map will create/update a Computer (cmdb_ci_computer) or Server (cmdb_ci_server, cmdb_ci_win_server, cmdb_aix_server etc..) based on the operating_system field value. The incoming computers are coalesced based on the serial_number field. Additionally, the transform script of the map will map to various extensions of the Computer (cmdb_ci_computer) based on the operating_system value being entered.

- UNIX Server (cmdb_ci_unix_server)
  - AIX
  - HP/UX
  - Solaris
  - AIX
- Windows Server (cmdb_ci_win_server)
  - Windows 2000 Server
  - Windows 2003 Server
  - Windows NT 4.0
- Server (cmdb_ci_server)
  - Any operating system that contains the word "Linux"
Table 355: imp_computer

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial_number</td>
<td>The unique identifier for this computer. It is marked as the coalescing value in the default transform map for the corresponding Computer and is mapped to the serial_number field of Computer (cmdb_ci_computer).</td>
<td>Character (40)</td>
</tr>
<tr>
<td>cpu_count</td>
<td>The number of CPUs that this computer has. It is mapped to the cpu_count field of the Computer (cmdb_ci_computer) table.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>cpu_speed</td>
<td>The clock speed of the CPU in MHz. This field is mapped to the cpu_speed field of Computer (cmdb_ci_computer)</td>
<td>Character (40)</td>
</tr>
<tr>
<td>cpu_type</td>
<td>Free form text describing the type of CPU. Example values are &quot;GenuineIntel&quot;, &quot;IBM&quot;, or &quot;Pentium 4&quot;. This field is mapped to the cpu_type field of Computer (cmdb_ci_computer)</td>
<td>Character (40)</td>
</tr>
<tr>
<td>disk_space</td>
<td>A numeric value describing the total disk space available to the computer in GB. This field is mapped to the disk_space field of Computer (cmdb_ci_computer)</td>
<td>Character (40)</td>
</tr>
<tr>
<td>manufacturer</td>
<td>A string name for the manufacturer of the computer. This field is mapped to the manufacturer field of Computer (cmdb_ci_computer) which is a reference to Company (core_company)</td>
<td>Character (40)</td>
</tr>
<tr>
<td>model_id</td>
<td>A string name for the model of the computer. This field is mapped to the model_id field of Computer (cmdb_ci_computer) which is a reference to Model Name (cmdb_model)</td>
<td>Character (40)</td>
</tr>
<tr>
<td>name</td>
<td>A string value representing the name of the Computer, usually a host name or IP/MAC address. It is mapped to the name field of Computer (cmdb_ci_computer)</td>
<td>Character (40)</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Data type</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>operating_system</td>
<td>A string value for the main operating system running on the computer. It is mapped to the os field of Computer (cmdb_ci_computer). Out of box values are:</td>
<td>Character (40)</td>
</tr>
<tr>
<td></td>
<td>• AIX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GNU/Linux</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• HP/UX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Linux Fedora</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Linux Red Hat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Linux SuSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Mac OS 10 (OS/X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Mac OS 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Mac OS 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Mac OS/X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• OS/400</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Solaris</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• SunOS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2000 Advanced Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2000 Datacenter Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2000 Professional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2000 Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2003 Datacenter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2003 Enterprise</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2003 Standard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2003 Web</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 98</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows ME</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows NT 4.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows XP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows XP Home</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows XP Professional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2000 Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 2003 Enterprise</td>
<td></td>
</tr>
<tr>
<td>ram</td>
<td>A numeric value for the total number of memory installed on this computer in MB. This value is mapped to the ram field of Computer (cmdb_ci_computer)</td>
<td>Character (40)</td>
</tr>
</tbody>
</table>
# User

A standard object for describing an external interface for a user in the system. The default transform map script sets the user_name field value to first_name.last_name if the web service's user_id field value is not supplied, otherwise, the user_id value is mapped directly to the user_name field in the User (sys_user) table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>A string value containing the user's email address. This value is mapped to</td>
<td>Character (40)</td>
</tr>
<tr>
<td></td>
<td>the email field in User (sys_user) and is set as the coalesce value for the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>transform.</td>
<td></td>
</tr>
<tr>
<td>department</td>
<td>The department the user is in. This field is mapped to the department field</td>
<td>Character (40)</td>
</tr>
<tr>
<td></td>
<td>in User (sys_user) which is a reference to the Department (cmn_department)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>table.</td>
<td></td>
</tr>
<tr>
<td>first_name</td>
<td>The first name of the user, mapped to the first_name field of the User</td>
<td>Character (40)</td>
</tr>
<tr>
<td></td>
<td>(sys_user) table.</td>
<td></td>
</tr>
<tr>
<td>last_name</td>
<td>The last name of the user, mapped to the last_name field of the User (sys_user) table.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>location</td>
<td>The location the user is in, mapped to the location field of the User (sys_user) table which is a reference field to Location (cmn_location)</td>
<td>Character (40)</td>
</tr>
<tr>
<td>phone</td>
<td>The phone number of the user, mapped to the phone (Business Phone) field of</td>
<td>Character (40)</td>
</tr>
<tr>
<td></td>
<td>the User (sys_user) table.</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>This is the user identification, usually a login name, that maps to the</td>
<td>Character (40)</td>
</tr>
<tr>
<td></td>
<td>user_name (User ID) field of the User (sys_user) table.</td>
<td></td>
</tr>
</tbody>
</table>

# Location

A standard object for describing an external interface for a location in the system. The web service will create or modify records in the Location (cmn_location) table.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the location, for example &quot;Headquarters&quot;, &quot;Sales office&quot; etc. This field is mapped to the name field of Location (cmn_location) and is part of the coalesce to search for an existing location.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>street</td>
<td>The street address of the location, for example &quot;1234 ServiceNow way&quot; etc. This field is mapped to the street field of Location (cmn_location) and is part of the coalesce to search for an existing location.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>city</td>
<td>The city of the location, for example &quot;San Diego&quot;, &quot;London&quot; etc. This field is mapped to the city field of Location (cmn_location) and is part of the coalesce to search for an existing location.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>state</td>
<td>The state of the location, for example &quot;California&quot;, &quot;Connecticut&quot; etc. This field is mapped to the city field of Location (cmn_location) and is part of the coalesce to search for an existing location.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>zip</td>
<td>The zip code for the location, for example &quot;92130&quot;, &quot;10001&quot; etc. This field is mapped to the zip field of Location (cmn_location) and is part of the coalesce to search for an existing location.</td>
<td>Character (40)</td>
</tr>
<tr>
<td>country</td>
<td>The country for the location, for example &quot;USA&quot;, &quot;United Kingdom&quot; etc. This field is mapped to the country field of Location (cmn_location).</td>
<td>Character (40)</td>
</tr>
</tbody>
</table>

Import sets maximum row size

Rows imported using import sets must not exceed the maximum row size.

A single row in a database may not contain more than 8126 bytes of data. The size of each row is determined by the amount of content in all fields, as well as the character set for text fields. For example, a row with 10 text fields each containing 1000 characters using a French character set takes 15360 bytes.
Attempting to import more data to a single row than the maximum size causes the import to skip that row. Any rows that were skipped for this reason are listed in the import log.

Importing data using import sets

An import set acts as a staging table to store raw data from an external source.

The import process uses a transform map to add or update data from the import set to an existing ServiceNow table such as incident or problem.

Use import sets to:
- Manually import data from a file on your local system
- Manually import data from a data source
- Periodically import data from a data source with a scheduled import
- Programmatically import data from a file with a script
- Programmatically import data from a web service

Figure 435: Importing data: data flow
Create a data source
Create a data source record to define what data an import set should import.

Role required: admin

1. Navigate to System Import Sets > Data Sources.
2. Click New.
3. Set the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for this data source.</td>
</tr>
<tr>
<td>Import set table label</td>
<td>Specify a label for the import set staging table to use.</td>
</tr>
<tr>
<td>Import set table name</td>
<td>ServiceNow uses the label you entered to construct a unique table name.</td>
</tr>
<tr>
<td></td>
<td>This prevents namespace collision with an already existing table.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the type of data you are importing, such as from a remote File, a JDBC</td>
</tr>
<tr>
<td></td>
<td>database, or using LDAP.</td>
</tr>
</tbody>
</table>

4. Set additional fields based on the selected Type.
   Refer to the documentation for each data source type for additional field information.

5. Click Submit.

Define a transform map for the new data source and run an import.

File type data source fields
These additional fields appear on the data source form when the Type field value is File.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Select one of the following formats from the list: CSV, CSV (tab), Excel, XML.</td>
</tr>
<tr>
<td>Zipped</td>
<td>Select this check box if the import file is compressed.</td>
</tr>
<tr>
<td>Xpath for each row</td>
<td>Specify an XPath expression that selects the nodes which become rows in the import table. The children of the selected nodes will become the columns in the rows.</td>
</tr>
<tr>
<td>Expand node children</td>
<td>Select this check box child elements of the node should be converted into additional columns. Clear this field if the parent column value should be an XML fragment.</td>
</tr>
<tr>
<td>File retrieval method</td>
<td>Choose the appropriate retrieval method for this file. See File retrieval methods on page 1513 for more information.</td>
</tr>
<tr>
<td>File path</td>
<td>Specify the path to the file to import.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SCP authentication method</td>
<td>Select to authenticate with a Username and Password or with a Public key.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot authenticate to data sources with a public-private key pair. Use a username and password authentication instead.</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>Enter the name of the server from which the file will be imported.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Enter the port to use to connect to the specified server. This field appears if you select HTTP or HTTPS as the file retrieval method.</td>
</tr>
<tr>
<td><strong>User name</strong></td>
<td>Enter the user name for authentication on the file server.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Enter the password for authentication on the file server.</td>
</tr>
<tr>
<td><strong>Private keyfile</strong></td>
<td>Specify the keyfile when using legacy SCP public-private keyfiles.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot authenticate to data sources with a public-private key pair. Use a username and password authentication instead.</td>
</tr>
<tr>
<td><strong>System KeyStore</strong></td>
<td>Select this check box to validate the certificate from the FTPS server against all saved certificates. This certificate may be any type supported by the instance. If you do not select this check box, the instance uses the Java default certificate to validate the FTPS server. This field is available only for data sources with a File retrieval method value of FTPS (Auth SSL)</td>
</tr>
</tbody>
</table>

**JDBC type data source fields**
These additional fields appear on the data source form when the Type field value is **JDBC**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use MID Server</strong></td>
<td>Select a MID Server to use to connect to this data source.</td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>Select one of the supported database formats. Instances can connect to Microsoft SQL Server 2012 but not Microsoft SQL Server 2000.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Instance name</td>
<td>Used to support dynamic port connectivity with SQL Server. To ensure connectivity, provide the instance name for a SQL Server that receives a new port number dynamically after each reboot. This field only appears when <strong>SQLServer</strong> is selected as a data source format.</td>
</tr>
<tr>
<td>Database name</td>
<td>Enter the name of the database instance.</td>
</tr>
<tr>
<td>Database port</td>
<td>Enter the port number for the database. Leave this field empty when using dynamic port connectivity with SQL Server.</td>
</tr>
<tr>
<td>Use integrated authentication</td>
<td>Select this check box to allow the JDBC connection to use the ID of the user configured for the Windows MID Server service for SQL Server authentication. For additional details, see <em>SQL server integrated authentication for Windows</em> on page 1521.</td>
</tr>
<tr>
<td>User name</td>
<td>Enter the user name for authentication on the database server. The user name provided for the JDBC connection must be associated with an account on that database server; it cannot be a Windows Domain account.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password for authentication on the database server.</td>
</tr>
<tr>
<td>Server</td>
<td>Enter the name of the server from which the data will be imported.</td>
</tr>
<tr>
<td>Query</td>
<td>Select the type of query to run: <strong>All rows from Table</strong> or <strong>Specific SQL</strong>. If you select to run a SQL statement, the SQL statement field appears.</td>
</tr>
<tr>
<td>Query timeout</td>
<td>Specify the number of seconds the JDBC driver will wait for a query (SELECT) to complete. Zero means no timeout. If timeout is exceeded, the integration considers the JDBC result inaccessible and places it in an error state.</td>
</tr>
<tr>
<td>Connection timeout</td>
<td>Specify the number of seconds before MID Server connection cache pool closes and removes it from the pool. Zero means no timeout.</td>
</tr>
<tr>
<td>Table name</td>
<td>Type the name of the table from which the data is being exported.</td>
</tr>
<tr>
<td>SQL statement</td>
<td>Type a SQL statement to extract the desired data from the database.</td>
</tr>
</tbody>
</table>
Geneva    ServiceNow    ServiceNow Platform

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use last run datetime</td>
<td>This selection controls the amount of data that is retrieved from a database during an import run. If this check box is unselected, then all rows in the table specified are imported, every time. You might want to use this setting if this is a one-time import, or if all the data in the target table is new. If this check box is selected, two additional fields appear, enabling you to select a datetime value to limit imported data to delta values only.</td>
</tr>
<tr>
<td>Last run datetime</td>
<td>The datetime value in this field is automatically populated from the database field you select below and represents the latest value from the previous run. This value acts as a dynamic filter to restrict the number of records retrieved to those records that have changed since the data source's last runtime.</td>
</tr>
<tr>
<td>Last run database field</td>
<td>Select the field name from the source table that will be used as the filter in the next run. This value may need to be case sensitive depending on the target database type.</td>
</tr>
<tr>
<td>Additional Database Parameters</td>
<td>Optional parameters to add to the end of the URL generated for this data source. Enter a semicolon-separated list of parameters. You may need to configure the form to view this field.</td>
</tr>
</tbody>
</table>

LDAP type data source fields
These additional fields appear on the data source form when the Type field value is LDAP.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP target</td>
<td>Select a target from the list of your LDAP OU definitions.</td>
</tr>
</tbody>
</table>

Create a transform map
Every import operation to a production table requires at least one transform map associated with an import set.

Role required: import_transformer, import_admin, or admin

The transform map specifies the data relationships between the import set and the target table. For every transformation, you must either create a new transform map or select an existing one.

1. Navigate to System Import Sets > Create Transform Map.
2. Fill in the fields, as appropriate (see table).

Table 359: Create transform map

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a user-friendly label for identifying the transform map.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Source table</td>
<td>Select the import table containing the raw import set data. An import table is any table that extends the Import Set Row [sys_import_set_row] table. You can select only tables within the currently selected application scope.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the transform map available for use.</td>
</tr>
<tr>
<td>Run Business Rules</td>
<td>Select this check box to run business rules, workflows, approval engines, auditing, and field normalization while the transformation inserts or updates data into the target table. Clearing this check box runs GlideRecord.setWorkflow() with a value of false.</td>
</tr>
<tr>
<td>Enforce Mandatory Fields</td>
<td>Choose whether to enforce mandatory fields on the target table.</td>
</tr>
<tr>
<td>Copy Empty Fields</td>
<td>Select this check box to clear fields with existing values when an incoming field contains an empty value. See Using NULL as a Field Value for information on how to import empty values.</td>
</tr>
<tr>
<td>Created</td>
<td>Shows the transform map creation date. This field is automatically populated.</td>
</tr>
<tr>
<td>Target table</td>
<td>Select the table where you want transformed data to be placed. You can select only tables within the currently selected application scope, the global scope, or tables that grant write access to other applications</td>
</tr>
<tr>
<td>Order</td>
<td>Enter the order in which to apply transform maps in the event that more than one map fits the conditions. ServiceNow runs transform maps from lowest to highest Order.</td>
</tr>
<tr>
<td>Run Script</td>
<td>Select this check box to display the Script field.</td>
</tr>
<tr>
<td>Script</td>
<td>Enter the transform map script you want to use to transform field values in the source table to the target table. ServiceNow runs the transform map script in addition to any Field Maps. By default, new transform map scripts are wrapped in immediately-invoked function expressions.</td>
</tr>
<tr>
<td>Field Maps</td>
<td>Use this related list to add one or more field maps. ServiceNow runs the transform Field Maps in addition to any transform script.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Transform scripts</td>
<td>Use this related list to add one or more transform map scripts. A transform script allows you to apply extra business logic at a specified stage of the transformation.</td>
</tr>
</tbody>
</table>

The sample User import transform map looks like this.
**Important:** The string `NULL` has a particular role in the code for the ServiceNow platform and is a reserved word. It should not be used as a field value in import set transform maps or anywhere in the First name or Last name fields. The reserved word is `NULL` in all capital letters. A field with the value `Null` or `null`, for example, is acceptable. `NULL` should be used only to clear out a particular field.

Create a field map

Field maps establish a relationship between a field in an import set table and a field in the target table.

Role required: import_transformer, import_admin, or admin

The field map determines what values from the source table the transformation adds to the target table.

1. Navigate to **System Import Sets > Transform Maps**.
2. Open a transform map.
3. In the **Field Maps** related list, click **New**.
4. Fill in the fields, as appropriate (see table).
5. Click **Save**.
6. Repeat steps 3-4 for each field mapping.
7. Click **Update**.

### Table 360: Creating a Field Map

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source table</td>
<td>Displays the table where the raw import set data is loaded (the source for the transformation). This field is automatically populated from the transform map. You can select only tables within the currently selected application scope.</td>
</tr>
<tr>
<td>Source field</td>
<td>Select the field on the source table to be transformed. This may be blank if the <strong>Source table</strong> only contains raw data.</td>
</tr>
<tr>
<td>Map</td>
<td>Displays the transform map that uses this field mapping. This field is automatically populated.</td>
</tr>
<tr>
<td>Date format</td>
<td>This field is available if the target field is a date or date-time field. This field specifies the date format of the source field.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Choice action         | This field is available if the target field is a choice list or reference field. This field specifies what to do if the import set contains a reference or choice value other than those available. Select one of these options:  
• create: Create a new choice or record in the reference table.  
• ignore: Ignore the new value from the source table.  
• reject: Skip the entire row (record) containing the new value and continue to the next row. |
| Use source script     | Select this check box to use a script instead of the Source field.  
Source script          | Enter a script to determine the source of this field mapping instead of the Source field. The script should return the answer variable. For example, this source script combines information from multiple sources into a single value. |
<p>| Target table          | Select the table where you want transformed data to be placed. You can select only tables within the currently selected application scope, the global scope, or tables that grant write access to other applications |
| Target field          | Select the field where the values from the source field is stored.                                                                     |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referenced value field name</td>
<td>When the target field is a reference field, the transform map needs a way to match incoming source values to existing records in the reference field's source table. Since most imports do not provide a 32-character sys_id value, you must specify a column from the reference field's source table that contains values that match the incoming source values. When there is a matching record, the transform map stores the sys_id of the matching record in the target field. If there is no matching record, the transform map creates a new record in the reference field's table and stores the sys_id of the new record in the target field. If you leave this field blank, the transform map looks for matching values from the display value column of the reference field table. For example, suppose you are importing incident records and the incoming data lists user IDs for the <strong>Assigned to</strong> field. If you leave <strong>Referenced value field name</strong> blank, the transform map searches for matching values in the User table's display value column: name. By setting the <strong>Referenced value field name</strong> to the user_name column you can match the user ID values to the appropriate user records.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Coalesce</td>
<td>Configuring a target field to coalesce causes the import set to treat the field as a unique key. When selected, the import set application attempts to match source values to records with values from an existing record. If a match is found, the transform map updates the record instead of creating a new record. When false, the import set application always creates new records for each transformation. If multiple fields are set to coalesce, all coalesce values must match an existing record. If two fields are set for coalescing and a matching value is found for one of the coalescing fields but not on the other, a new record is inserted. You must create an index on the target table if none of the coalesce fields are indexed. Indexing can improve performance for read and write operations. If one or more coalesce fields already has an index, you do not need to create an additional index. After setting the Coalesce value for all fields on the target table, use the Index Coalesce Fields related link to create an index. Before creating the index, ensure that the Coalesce value is set correctly for all fields on the target table to avoid creating unnecessary indexes for the same target table.</td>
</tr>
<tr>
<td>Coalesce empty fields</td>
<td>Select this check box to match an empty source field value to an empty target field value. For example, the User transform map coalesces on the email field. With this option selected, a source record containing an empty email address coalesces to a target record containing an empty email address.</td>
</tr>
<tr>
<td>Coalesce case sensitive</td>
<td>Select this check box to have case sensitive coalesce values result in the creation of new records. By default, values marked as Coalesce are used in a case insensitive lookup for existing records. Case insensitive records only update existing records and do not cause the creation of new records.</td>
</tr>
</tbody>
</table>

A completed field map record with coalescence enabled might look like this:
Field map script variables
Several predefined variables are available in the Source script field.

Table 361: Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The record from the source table that is currently being processed.</td>
</tr>
<tr>
<td>target</td>
<td>GlideRecord</td>
<td>The record from the target table that is currently being processed.</td>
</tr>
<tr>
<td>answer</td>
<td>String</td>
<td>The field value to set in the target record.</td>
</tr>
<tr>
<td>map</td>
<td>GlideTransformMap</td>
<td>Read-only information about the current transform map record.</td>
</tr>
<tr>
<td>log</td>
<td>Function</td>
<td>The log object for the current import run. Use this object to log messages such as log.info(&quot;&lt;Message&gt;&quot;) or log.warn(&quot;&lt;Message&gt;&quot;). This object logs to the import log for standard import sets, or other log locations as required, such as for web service imports.</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>Contains either the value insert or update depending on whether the current target row will be created or updated.</td>
</tr>
</tbody>
</table>

Mapping options
You can map fields in a number of ways depending on the circumstances of the import and whether data must be transformed prior to loading onto a production table.

It is also important to note that any given import operation may require taking advantage of multiple mapping methods, and these methods can be readily used in conjunction with one another.

**Automatic Mapping Utility**

The simplest mapping method is where all of the field names of the import sets match the names of the fields on the production tables onto which the data will be transformed. In this case, simply click **Auto map matching fields** in the related list in the Table Transform Map form and confirm proper matching. If there are any discrepancies in terms of how fields were automatically matched, use the mapping assist utility to correct them. When all fields are matched properly, click the **Transform** related link to begin transforming data onto the destination table.

**Mapping Assist Utility**

The mapping assist utility provides a visually intuitive environment for specifying mapping between import set fields and production table fields. The mapping assist utility makes it possible to map a single source field (field on an import set table) to multiple destination fields (fields on a production table).
Figure 436: Mapping assist

Changing the Date Format

If the date format of the source field does not match the format of the target field, you can set a date format mapping to transform dates from one format to another. For example, this mapping specifies that the date format of the CSV source file uses the **MM-dd-yyyy** format.
Date format options include:

Table 362: Mapping Options

<table>
<thead>
<tr>
<th>Date Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd-MM-yyyy</td>
<td>Day-month-year</td>
</tr>
<tr>
<td>dd-MM-yyyy hh:mm:ss</td>
<td>Day-month-year hours-minutes-seconds</td>
</tr>
<tr>
<td>dd-MM-yyyy hh:mm:ss z</td>
<td>Day-month-year hours-minutes-seconds timezone</td>
</tr>
<tr>
<td>MM-dd-yyyy</td>
<td>Month-day-year</td>
</tr>
<tr>
<td>yyyy-MM-dd</td>
<td>Month-day-year hours-minutes-seconds</td>
</tr>
<tr>
<td>yyyy-MM-dd hh:mm:ss</td>
<td></td>
</tr>
</tbody>
</table>
### Date Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-dd-yyyy hh:mm:ss z</td>
<td>Month-day-year hours-minutes-seconds timezone</td>
</tr>
<tr>
<td>yyyy-dd-MM</td>
<td>Year-day-month</td>
</tr>
<tr>
<td>yyyy-MM-dd</td>
<td>Year-month-day</td>
</tr>
<tr>
<td>yyyy-MM-dd hh:mm:ss</td>
<td>Year-month-day hours-minutes-seconds</td>
</tr>
</tbody>
</table>

You can specify a custom date format using **HH** to denote 24 hour time. Converting from a 24 hour to 12 hour date format may cause times from 12:00 to 12:59 to be formatted as 00:00.

### Mapping to a Duration Field

ServiceNow duration fields use a special data type that lists the number of milliseconds the duration value is. To map import data to a duration field use one of the following methods to transform source values into a duration.

- Calculate the duration from a start and end date
- Convert an existing duration value into a ServiceNow duration value

### Calculating a Duration Value from a Start and End Date

If the import source has a start and end date, you can calculate a duration with JavaScript.

1. Navigate to **System Import Sets > Transform Maps**.
2. Select the transform map you want to calculate a duration value. For example, the Notification transform map that imports into the Incident table.
3. Select the **Run script** check box.
4. Enter JavaScript to transform the start and end dates into a duration. See the sample script.
5. Click **Update**.

This sample script transforms the source.u_start and source.u_end fields to a duration value in the target.duration field. Change the field names to match your source and target fields as needed.

```javascript
target.duration = gs.dateDiff(source.u_start.getDisplayValue(), source.u_end.getDisplayValue(), false);
```

### Convert a Duration Value into a ServiceNow Duration Value

If the import source already contains a start date and a duration value, you can convert the existing duration into a ServiceNow duration. For example, you might have an Excel data source that lists a duration in seconds. ServiceNow expects durations to have millisecond values.

1. Navigate to **System Import Sets > Transform Maps**.
2. Select a transform map.
   - For example, the Notification transform map that imports into the Incident table.
3. Select the **Run script** check box.
4. Enter JavaScript to convert existing duration values into a ServiceNow duration values. See the sample script.

5. Click Update.

This sample script converts a duration in seconds (from the source.u_duration field) to a duration in milliseconds (in the target.duration field). Change the field names to match your source and target fields, as needed.

```javascript
//Transform the value in source.u_duration from seconds to milliseconds
target.duration.setDateNumericValue(source.u_duration * 1000);
```

This sample script converts a duration already in milliseconds (from the source.u_duration field) to a ServiceNow duration (in the target.duration field). Change the field names to match your source and target fields, as needed.

```javascript
//Transform the value in source.u_duration to ServiceNow format
target.duration.setDateNumericValue(source.u_duration);
```

### Using a Script to Calculate a Source Value

You may want to use a source script instead of the Source field when:

- The source value is not in the format for the mapped target value.
- You want to look up a value before mapping to the target field.
- The source value must be computed from multiple fields.
- You need to create a compounded or calculated coalesce value in the target field.

For example, this example source script computes the value of the user_name field when it is blank in the source.
A source script expects the calculated value to be set in the global variable `answer`.

Mapping Binary or BLOB Fields

ServiceNow uses a special process to import binary and binary large object (BLOB) data from JDBC data sources. All binary data is automatically converted into a record in the Attachment [sys_attachment] table before the transformation occurs. The import set table only stores the attachment record's sys_id value in the import table field instead of the actual binary value. For example, suppose you use a JDBC data source to import data from a CA Service Desk system, which stores each record's key value as binary data. When you import the Service Desk key values into a ServiceNow table, the ServiceNow field only contains a sys_id reference to the corresponding binary data in the Attachment Record [sys_attachment] table rather than the actual binary data.

To have a transform map access the attachment, use the GlideRecord API in an onAfter script. The script needs to run in an onAfter event because the target.sys_id object is only available after the data is placed in the import set table. For example, to map the resulting attachment to the target transform record, you can use the following script.

```javascript
var agr = new GlideRecord("sys_attachment");
agr.addQuery("sys_id", source.u_blob_field);
// the source field needs to be mapped to the source that is the BLOB
agr.query();
if(agr.next()) {
    agr.table_name = "cmdb_ci"; // the target table of the transform map
    agr.table_sys_id = target.sys_id; // the target record
    agr.content_type = source.u_contenttype;
```
If you are mapping directly to the `db_image` table, run the following `onAfter` script to display the image:

```javascript
var strTemp = source.u_file_name;
var fType = strTemp.substr(- 3);
var iCont = "";
if(fType == 'jpg') {
    iCont = 'jpeg';
} else {
    iCont = fType;
}

var agr = new GlideRecord("sys_attachment");
agr.addQuery("sys_id", source.u_blob_field);
agr.query();
if(agr.next()) {
    agr.table_name = "ZZ_YYdb_image";
    agr.table_sys_id = target.sys_id;
    agr.content_type = 'image/' + iCont;
    agr.file_name = 'image';
    agr.update();
}
```

**Run an import**

You can manually run an import to immediately import data.

1. From a Transform Map, click **Run Import**.
   When the import is done, you'll see a link to go straight to the target table containing your imported records. The amount of time that it takes to run an import varies depending on the number of record to be imported and may take as long as several hours for very large import operations (tens of thousands of records).

2. [optional] Click on the link **View the imported data** to see the loaded import set table.

3. [optional] Click on the link **Create transform map** to create a new transform map to transform the data in the import set table to its target table.

4. [optional] Click on the link **Run import** to execute an existing transform map for the loaded data.

Three things to note at this point:
- The spreadsheet was imported, and a new table was created to hold the data.
- Within that table, the imported records are designated with their own "Set" value.
- A new module was created in the System Import Sets application for the new table.

**Cancel an import in progress**

You can stop an import that is taking too much time.

1. Navigate to **System Import Sets > Advanced > Progress**.
2. Open the Progress Worker that must be canceled.
3. Click the **Cancel job** related link.
Review the import set
Even if no errors were reported by the progress indicator, review how your data was imported into the new import set.

1. Click **View imported data**.
   
   There are several fields that will be visible in the table that were not part of the original data source, these are system rows can be used to facilitate scripted import operations.

2. Review the data.

3. Return to the previous page.

Create a transform map to associate with this import set.
Control import table dictionary entries
  
  Control whether an import can automatically change the dictionary entries for staging tables.
  
  - Set the system property to false to prevent an import from changing a staging table's dictionary entry.
### glide.import_set_row.dynamically_add_fields

Specifies whether an import set can add new columns to the staging table (true) or not (false). Instances that contain large numbers of import sets can sometimes become unresponsive when an import adds a column because the instance must alter every row in the staging table. In some cases, the database alter table action causes an outage. Setting this property to false prevents an import set from adding columns to the staging table and produces a log message. As a workaround, administrators can manually add a column to the staging table by creating a new dictionary entry and then reimporting the import set.

- **Type:** true | false
- **Default value:** true
- **Location:** Add to the System Properties [sys_properties] table

### com.glide.import_set.importlog_level

Specifies how much information import sets add to the log. Possible values are INFO, WARNING, and ERROR. You can set this to INFO in dev environment and change it to WARNING or ERROR in production to reduce amount of logging noise traffic to the database.

- **Type:** string
- **Default value:** INFO
- **Location:** Add to the System Properties [sys_properties] table

---

**Set the import log level**

Set the import log level to view detailed log information about import sets.

- Use the system property to control the log level for import sets.

---

#### Schedule a data import

Scheduled imports make it is possible to specify that certain import operations should occur at some regular interval.

**Role required:** import_scheduler, import_admin, or admin

**Important:** Do not schedule multiple imports at the same time. Running multiple imports concurrently may negatively impact performance or cause an instance outage.

1. Navigate to **System Import Sets > Administration > Scheduled Imports.**
2. Complete the form (see table).

Table 363: Scheduled Data Import form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for this scheduled data import.</td>
</tr>
<tr>
<td>Data source</td>
<td>Select the data source record that defines the data to import.</td>
</tr>
<tr>
<td></td>
<td>If you select a remote datasource, it is refreshed via the remote connection prior to the import operation.</td>
</tr>
<tr>
<td>Run as</td>
<td>Enter the name of a user whose credentials the import job will use. If you do not specify a value, the import job runs as the guest user.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to activate the scheduled import.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Run</td>
<td>Select the frequency at which you want to run the import.</td>
</tr>
<tr>
<td>Time</td>
<td>Enter the time you want the import to begin. Do not schedule multiple imports at the same time. Running multiple imports concurrently may negatively impact performance or cause an instance outage.</td>
</tr>
<tr>
<td>Conditional</td>
<td>Select this check box if you want to write a script to define the conditions that must evaluate to true before the import will run.</td>
</tr>
<tr>
<td>Condition</td>
<td>Write the script to be used to evaluate whether an import should run. This field is visible if you selected the Conditional check box.</td>
</tr>
<tr>
<td>Repeat interval</td>
<td>Select the period of time that must elapse before the import will be repeated. This field is visible if Run is set to Periodically.</td>
</tr>
<tr>
<td>Starting</td>
<td>Click the calendar icon and select the date on which the import should begin. This field is visible if Run is set to Periodically.</td>
</tr>
<tr>
<td>Execute pre-import script</td>
<td>Select this check box if you want to write a script to be run before the import is performed.</td>
</tr>
<tr>
<td>Pre script</td>
<td>Write the script you want to run before the data is imported. This field is visible if you selected the Execute pre-import script check box.</td>
</tr>
<tr>
<td>Execute post-import script</td>
<td>Select this check box if you want to write a script to be run after the import is performed.</td>
</tr>
<tr>
<td>Post script</td>
<td>Write the script you want to run after the import finishes. This field is visible if you selected the Execute post-import script check box.</td>
</tr>
</tbody>
</table>

Data import scripting options
Scheduled Import Pre and Post scripts have access to multiple JavaScript objects.
Table 364: Data import scripting options

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancel</td>
<td>Set this object to true to stop the import action.</td>
<td>Use the Pre script field to evaluate the conditions of the import and determine whether to cancel the import process. To cancel the import process, use the following call:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cancel = true;</td>
</tr>
<tr>
<td>glide.scheduled_import.stop_on_error</td>
<td>Set this object to true to stop the import process when the parent scheduled import generates an error.</td>
<td>One import set can depend on the results of another import set. Use the following code to stop the current import set when the parent import set generates an error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>glide.scheduled_import.stop_on_error = true;</td>
</tr>
<tr>
<td>import_set</td>
<td>Get the GlideRecord object for the new import set. This variable allows you to query the following columns from the sys_import_set table:</td>
<td>If you want to use information from the import set, you can specify one of the properties of the import_set variable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>var x = import_set.number;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_source</td>
<td>Specify the data source you want the scheduled import to use. The data source must be an existing GlideRecord object.</td>
<td>Typically, you define the data source with the Scheduled Data Import record. If you want to override this data source and use another data source in certain conditions, you can use the following call:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>data_source = &quot;Example LDAP Users&quot;;</td>
</tr>
</tbody>
</table>

*Post CSV or Excel files directly to an import set*

An administrator can post CSV or Excel files directly to instance.

The sys_import.do target lets you dynamically upload a CSV or Excel file into the import set table specified by the sysparm_import_set_tablename parameter. You must specify a name that matches an existing import set table name, if it does not exist, pre-create it by doing a manual import. The sysparm_transform_after_load=true parameter makes the CSV transform execute immediately, if a transform map exists.
• Use the following post syntax: https://instance.service-now.com/sys_import.do?sysparm_import_set_tablename=table_name&sysparm_transform_after_load=true&uploadfile=path/to/file.csv

Preserve leading spaces in excel imports
You can preserve leading spaces when importing data from Excel.
Role required: admin

Attention: The import process always removes trailing spaces from Excel data cells.

• Use the system property `glide.import_set.preserve.leading.spaces` to preserve leading spaces when importing data from Excel.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.import_set.preserve.leading.spaces</td>
<td>Specifies whether the import process preserves leading spaces in Excel data cells. When false, the import process removes leading spaces from Excel data cells. When true, the import process preserves leading spaces.</td>
</tr>
</tbody>
</table>
  * Type: true | false
  * Default value: false
  * Location: Add a property using `sys_properties.list` on page 1411 to the System Properties [sys_properties] table

Ignore bad data rows in CSV files
Ignore data from rows in CSV files that the instance cannot parse.
Role required: admin
By default, imports cannot ignore bad data rows in CSV files and fail on the first error.
• To allow an import to ignore one or more rows of bad data, set the system property `com.glide.csv.loader.ignore_non_parseable_lines` to true.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.glide.csv.loader.ignore_non_parseable_lines</td>
<td>Allows an instance to ignore one or more lines (rows) that contain bad data (such as a row that is missing a column of data).</td>
</tr>
</tbody>
</table>
  * Type: true | false
  * Default value: false
  * Location: Add the property

• To specify the maximum number lines (rows) that an instance can ignore before the import fails, set the system property `com.glide.csv.loader.max_errors_allowed`. 
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| com.glide.csv.loader.max_errors_allowed | Specify the maximum number of lines (rows) that an import can ignore before failing. If the import succeeds, the import lists the number of rows the import ignored due to errors.  
- Type: integer  
- Default value: 100  
- Location: Add the property |

Posting a CVS file - Perl example
An example using Perl to post a CSV file with basic auth credentials.

```perl
# file: uploadafile.pl  
# call me like this:  
# uploadafile.pl --url="https://instance.service-now.com/sys_import.do?sysparm_import_set_tablename=dloo_test&sysparm_transform_after_load=true"  
# --uploadfile=/Users/davidloo/Desktop/test_files/test_users.csv  
#  
# the "sysparm_transform_after_load=true" parameter instructs the import set to transform immediately after loading  
#  
use strict;  
use LWP::UserAgent;  
use HTTP::Request::Common;  
use Getopt::Long;  
use File::Basename;  

my ( $o_url, $o_fqn );  
GetOptions(  
   "url=s" => \$o_url,  
   "uploadfile=s" => \$o_fqn,  
);  

# mandatory arguments: url  
&usage unless ( \$o_url && \$o_fqn );  
my \$url = \$o_url;  
my \$fname = \$o_fqn;  

# put timeouts, proxy etc into the useragent if needed  
my \$ua = LWP::UserAgent->new();  

# setup basic authentication credentials  
\$ua->credentials(  
   'demo.service-now.com:443',  
   'Service-now',  
   'admin' => 'admin'  
);  

my \$req = POST \$url, Content_Type => 'form-data',  
   Content => [  
      submit => 1,  
      upfile => [ \$fname ]  
   ];  
my \$response = \$ua->request(\$req);```

© 2017 ServiceNow. All rights reserved. 1578
if ($response->is_success()) {
    print "OK: ", $response->content;
} else {
    print $response->as_string;
}
exit;

sub usage {
    printf "usage:&nbsp;%s --url=%s --uploadfile=%s\n",
        basename(50), 'https://....', 'c:/data/test.csv';
    exit
}

Posting a CSV file - Java example

An example using the Java Apache HttpClient class to post a CSV file with basic auth credentials.

**Attention:** The Apache HttpClient may limit the amount of data you can import in a single transaction. This example is meant as a starting point and should not be used in production.

```
HttpClient httpclient = new HttpClient();
PostMethod post = new PostMethod("https://instance-name.service-now.com/
    sys_import.do?
    sysparm_import_set_tablename=u_test_upload&sysparm_transform_after_load=true");
try {
    Credentials defaultcreds = new UsernamePasswordCredentials("admin",
        "admin");
    httpclient.getState().setCredentials(AuthScope.ANY, defaultcreds); // Prepare HTTP post
    httpclient.getParams().setAuthenticationPreemptive(true);
    File targetFile = new File("/Users/davidloo/Desktop/test_files/
        nodeinfo2736820198834983863.csv");
    Part[] parts = { new FilePart(targetFile.getName(), targetFile) };
    post.setRequestHeader(new MultipartRequestEntity(parts,
        post.getParams()));
    int result = httpclient.executeMethod(post);
    // Display status code
    System.out.println("Response status code: " + result);
    // Display response
    System.out.println("Response body: "+post.getResponseBodyAsString());
} catch(Exception e) {  
    System.err.println(e.getMessage());
} finally {
    // Release current connection to the connection pool
    // once you are done
    post.releaseConnection();
}
```

**Completed import sets**

After an import set completes, you can review the completed import and clean up import set tables.

**Viewing the import log**

The import log is where you can find information about the internal processing that occurs during an import operation.
The log includes information generated by the system during the every step of the import operation from initialization of the import set table from a data source to transformation onto a target table. It is also possible to customize log outputs during the transformation by using log.info, log.warn, log.error statements.

**Figure 440: Import Log**

<table>
<thead>
<tr>
<th>Created</th>
<th>Level</th>
<th>Message</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Using ExcelLoader2</td>
<td>DataSource</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>getAsFile: attachment: %sys_data_source\abc\def\ghi\jkl\mno\pqr\stuv\wxyz\user\ids</td>
<td>DataSource</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Using attachment handler</td>
<td>DataSource</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Using tempFile: C:\Program Files\Apache Software Foundation\Tomcat 5.0.1\temp\ids53719.tmp</td>
<td>DataSource</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Loading POP\FileSystem: C:\Program Files\Apache Software Foundation\Tomcat 5.0.1\temp\ids53719.tmp</td>
<td>ExcelLoader</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Loading workbook</td>
<td>ExcelLoader</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Processing sheet Page 1 from workbook</td>
<td>ExcelLoader</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Created new table: u_user_import</td>
<td>ExcelLoader</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Processed: 0</td>
<td>ExcelLoader</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Processed: 6, inserts 6, updates 0, errors 0, empty and ignored 0</td>
<td>ExcelLoader</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Using import set: RSET10001, table u_user_import</td>
<td>ImportSetTransformer</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>total: 6, inserts 6, updates 0, errors 0, skipped 0, errors 0</td>
<td>ImportSetTransformer</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Using JDriver</td>
<td>DataSource</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>getAsFile: attachment: %sys_data_source\abc\def\ghi\jkl\mno\pqr\stuv\wxyz\user\ids</td>
<td>DataSource</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Using tempFile: C:\Program Files\Apache Software Foundation\Tomcat 5.0.1\temp\ids53721.cvs</td>
<td>DataSource</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Setting JDBC driver: org.apache.jdbc.Driver</td>
<td>Loader (test_users_tab.csv)</td>
</tr>
<tr>
<td>2023-03-13</td>
<td>Information</td>
<td>Connection path: jdbc:sqlite: csv: C:\Program Files\Apache Software Foundation\Tomcat 5.0.1\temp</td>
<td>Loader (test_users_tab.csv)</td>
</tr>
</tbody>
</table>

**Note:** Functionality described here requires the *Admin* role.
Run import utility

The Run Import utility is used to run an import operation using an existing Transform Map and Import Set table.

**Figure 441: Import History Record**

Import set scheduled cleanup

Import set scheduled cleanup is a scheduled job that runs every day at midnight.

By default, it will query all import sets that are older than 7 days and delete the import set and all related tables, eg. the import set table records. If this scheduled job is not active on your instance, we recommend that you activate it as soon as possible to mitigate the growing import set tables that you may have, especially if they are recurring imports.

**Note:** If the scheduled cleanup is not currently active on your system, you must ensure that your import set tables are not too big (> 5 million records total across all tables) before enabling this cleanup. If the tables are too big, you have to manually delete the records first before proceeding - please contact ServiceNow support to coordinate the deletion of this data.
Web service import sets

Web Service Import Sets complement Direct Web Services and Scripted Web Services in providing a web service interface to Import Set tables.

![Diagram of SOAP insert, Import Set Table, Transform, Target Table](image.png)

**Figure 442: WS Iset**

This type of web service will transform the incoming data synchronously based on the associated transform maps by default. If the associated import set mode is set to Asynchronous, the behavior is to save the...
data for transformation at a later time. Web Service Import Sets tables publish all the default Web Service functions in the WSDL.

![System Web Services](image)

Figure 443: System Web Services

This plugin also provides the following standard import set tables:

- Computer
- Location
- Notification
- User

Web service import set WSDL are accessed by specifying the import set table name + ".do?WSDL" on the URL. For example:

http://<instance name>.service-now.com/imp_notification.do?WSDL (The System Web Service plugin must be enabled first)

Creating an import set web service

Create a new web service import set table to define how to stage and transform imported data.

Navigate to System Web Services > Create New.
The Name of the web service is the table name of the import set table whereas the Label field is the resulting table field.

If you want to create a transform map after creating the web service, check the Create transform map checkbox and choose the target table you want the data to transform into. After the Create button is clicked, the web service will be created and you will be immediately put into the Table Transform Map form. You may then continue to specify the transform map or script.

**Web Service Fields**

The fields available for this web service. All fields by default are published as the XSD type of xsd:string. The Name is the field that is exposed for the web service and therefore appears as the name of the field in the WSDL. The Label is the label of the field as it appears for the import sets table.

You can Add, mark for Delete or modify (double click the field) an existing web service field in this list.

**Note:** After adding web service fields, click Create to create the web service import set table.

To add other fields after the Web Service is created, find the target table, and add the fields to that table.

**Mapping web service import sets**

During the creation of the web service import set, you may optionally create the transform map for it.
All transform maps will be executed for the service when it is invoked if the import set mode is set as "Synchronous" (the default).

The following image is an example of the transform map associated with the Notification web service import set.
Figure 445: Soap Transform Map

Script:

```java
target.comments = "Timestamp: " + source.timestamp + 
"\nExpires on: " + source.expires_on + 
"\nDuration: " + source.duration;
```
Adding Web Service Response Values

In the transform map script associated with a web service import set, setting certain variable values have the effect of changing the response values of the web service. In addition to the normal variables that are available in a transform map script, the table below documents the variables that are available and their effects.

Table 365: Adding Web Service Response Values

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>Output Object</td>
<td>Javascript object that holds dynamically created response elements used to customize the output response of a web service import set insert.</td>
</tr>
</tbody>
</table>

Example

```java
// create new elements called "transaction_id"
// and "hello" in the web service response
response.transaction_id="abc123";
response.hello="world";
status_message="message 1";
// this is the normal status_message variable
```

The code snippet above results in the following response being generated back to the web service consumer

```xml
<soapenv:Envelope
  xmlns:imp="http://www.service-now.com/imp_notification"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <insertResponse
      xmlns="http://www.service-now.com/imp_notification">
      <sys_id>969d157c0a0a0baf008ba5770ffa798c</sys_id>
      <table>incident</table>
      <display_name>number</display_name>
      <display_value>INC0010091</display_value>
      <status>Inserted</status>
      <status_message>message 1</status_message>
      <transaction_id>abc123</transaction_id>
      <hello>world</hello>
    </insertResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Debugging web service import sets

If you need to debug a SOAP Request coming into the system, create the system property `glide.processor.debug.SOAPProcessor`.

Once you have created it, set it to true to have all SOAP requests be logged in the System Log. Set it to false when you are done so as to keep the size of your System Log to a managed length.

Web service import set mode

When a SOAP message inserts a record into an import set table, and the table did not previously have an import set referenced, a new one will be created, and its Mode will be set as Synchronous.
An import set with a Mode of Synchronous will transform the data as soon as it is inserted (provided that the transform map already exists). This import set will also have a default State of Loading. By default, all Synchronous import sets will automatically be modified to Processed at midnight. As a result, when a new insert happens to the same table, a new Synchronous import set will be created.

![Figure 446: Synchronous import set](image)

Changing this import set to a mode of Asynchronous and a state of Loading has the effect of not transforming the incoming data as it is inserted, but rather "loading" the import set and deferring the data transformation later, either manually, or with a scheduled script job.

<table>
<thead>
<tr>
<th>Mode</th>
<th>State</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous</td>
<td>Loading</td>
<td>Data transformation is not occurring automatically and immediately. Data added to import set row has a state of &quot;Pending&quot;. Transform can be scheduled or executed manually when state is changed to Loaded</td>
</tr>
<tr>
<td>Asynchronous</td>
<td>Loaded</td>
<td>Marks the completion of data loading. Data transformation can now occur in a scheduled fashion or manually.</td>
</tr>
<tr>
<td>Synchronous</td>
<td>Loading</td>
<td>Data transformation is occurring automatically and immediately whenever data is inserted into the associated import set row.</td>
</tr>
</tbody>
</table>
### Controlling Insert Behavior

In imports sets that specify one or more coalesce fields, records with a matching coalesce value are transformed from source to target table serially (one at a time) to prevent duplicates.

In import sets that do not specify any coalesce field, records are transformed concurrently. You can control this behavior using the `glide.import_set_insert_serialized_when_no_coalesce` property.

The system property `glide.soap.import_set_insert_serialized.<table name>`, controls how the instance inserts records from web service calls into a specific import set table. When true, this property prevents identical simultaneous inserts from creating duplicate records by serializing the database insert operations. If a target table does not have any coalesce fields defined in a transform map, set this property to false to improve web service import set performance. The `glide.import_set_insert_serialized.<table name>` provides the same functionality.

Note: Setting this property to false can result in the creation of duplicate records.

### Standard SOAP Response

The SOAP response from a web service import set insert call returns the following standard values:

```
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
<SOAP-ENV:Body><insertResponse><sys_id>fa648f5f0a0a0b2b0048e7012448b8f1</sys_id><table>incident</table><display_name>number</display_name><display_value>INC10014</display_value><status>inserted</status></insertResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys_id</td>
<td>The Sys_id of the resulting record that was created or modified</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>table</td>
<td>The table name of the table that was affected. In the case of an Asynchronous call, the table name would be the import set table eg. imp_notification for the Notifications web service import set table</td>
</tr>
<tr>
<td>display_name</td>
<td>The name of the field that is set as the display field for the record that was created or modified</td>
</tr>
<tr>
<td>display_value</td>
<td>The value of the field designated as the display field. For example, the display field for the Incident table is the Number field and an example value would be INC10001</td>
</tr>
</tbody>
</table>
| status              | A string value that indicates the action that occurred as a result of the web service invocation, relating to the record defined by the sys_id and table field values  
  • inserted - the record was inserted  
  • updated - the record was updated  
  • ignored - the input was ignored, the record was not updated and no new record was created  
  • skipped - the input data was skipped (similar to ignored) due to missing coalesce values  
  • error - there was an error processing the input                                                                                               |
| status_message      | This value translates to the value found in the Comment field of the import set row and usually contains information related to the status value eg. "No field values changed" when the status is "ignored". Setting this value to a customized string value will cause the SOAP response to contain an optional status_message field to be returned. |
| error_message       | The message related to a status of error. When an error occurs, setting this value to a customized string value will cause the SOAP response to contain an optional error_message field to be returned. |

**Tailoring the SOAP Response**

It is possible to include information other than the information specified in the WSDL by overwriting the contents of status_message using the transform script.

*Inserting multiple records using insertMultiple*

You can insert multiple records in one SOAP request by using the insertMultiple operation.

The insertMultiple operation is available for the Direct Web Service API and Web Service Import Sets. To enable insertMultiple, activate the Insert Multiple Web Service plugin.
**Note:** Activating this plugin adds a new operation to the WSDL. After this plugin is activated, consume a new WSDL to update your web services client.

**Web service import sets security requirements**

Web Service Import Sets use the same security mechanisms as SOAP Web Services.

- Basic authentication requires a Web Service user provide a valid user name and password
- Contextual security requires a Web Service user meet the queried table's access control rules

If your instance uses high security settings, the Web Service user may also need the soap role.

**Web service import sets related links**

When displaying a mapped web service table, you will have the following related links.

- Import Sets - the import sets related to this web service import set
- Transform Maps - a list of transform maps related to this web service
- Transform History - the transformation history
- Edit Web Service - edit the web service

The following image shows a record that was inserted into the web service import set Notification. The target record is the resulting creation or modification to the Incident table record as a result of the transform.

![Figure 447: Soap Notification](image)

**Figure 447: Soap Notification**

**Web service import sets examples**

This example demonstrates the WSDL, SOAP envelope and response, Perl invocation, and result of a SOAP web service import.
Sample WSDL

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
    <wsdl:types>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified" targetNamespace="http://www.service-now.com/imp_notification">
            <xsd:element name="insert">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element maxOccurs="1" minOccurs="0" name="corrective_message" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="duration" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="expires_on" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="message" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="severity" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="source" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="timestamp" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="type" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="1" name="uuid" type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="insertResponse">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element maxOccurs="1" minOccurs="1" name="sys_id" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="1" name="table" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="1" name="display_name" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="1" name="display_value" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="1" name="status" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="status_message" type="xsd:string"/>
                        <xsd:element maxOccurs="1" minOccurs="0" name="error_message" type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </wsdl:types>
    <wsdl:message name="insertSoapOut">
        <wsdl:part name="imp_notification" element="tns:insertResponse"/>
    </wsdl:message>
    <wsdl:message name="insertSoapIn">
        <wsdl:part name="imp_notification" element="tns:insert"/>
    </wsdl:message>
    <wsdl:operation name="insert">
        <wsdl:input message="sncns:insertSoapIn"/>
        <wsdl:output message="sncns:insertSoapOut"/>
    </wsdl:operation>
    <wsdl:binding name="ServiceNowSoap" type="sncns:ServiceNowSoap">soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="insert">
        <soap:body use="literal"/>
    </wsdl:operation>
</wsdl:service>
</wsdl:definitions>
```

Sample SOAP Envelope

```xml
<?xml version="1.0" encoding="UTF-8"?>
    <SOAP-ENV:Body>
        <imp_notification:insert>
            <corrective_message></corrective_message>
            <duration></duration>
            <expires_on></expires_on>
            <message></message>
            <severity></severity>
            <source></source>
            <timestamp></timestamp>
            <type></type>
            <uuid></uuid>
        </imp_notification:insert>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Sample SOAP Response

<?xml version="1.0" encoding="UTF-8"?>
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:SOAP-ENC2="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:SOAP-ENV2="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <insertResponse>
      <sys_id>b54aafbc0a8006f0058db95daa5b88d</sys_id>
      <table>incident</table>
      <display_name>number</display_name>
      <display_value>INC10008</display_value>
      <status>ignored</status>
      <status_message>No field values changed</status_message>
    </insertResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Example Invocation using Perl

The following example script will use the Notification web service to create an Incident as the itil user. It uses the Perl language and the SOAP::Lite package.

```perl
#!/usr/bin/perl -w

#use SOAP::Lite ( +trace => all, maptype => {} );use SOAP::Lite;

sub SOAP::Transport::HTTP::Client::get_basic_credentials{return 'itil' =>'itil';// set basic auth credentials for the itil user}

my $soap= SOAP::Lite->proxy('http://localhost:8080/glide/imp_notification.do? SOAP');

my $method= SOAP::Data->name('insert')->{xmlns =>'http://www.service-now.com/'};

# insert into the web servicemy@params=( SOAP::Data->name(message =>'problem
detected for database DB12DG'));push(@params, SOAP::Data->name(source =>'DB12DG'));push(@params, SOAP::Data->name(uuid =>'HGAF76251HGF2'));

my $result=$soap->call($method=>@params);

print_result($result); //print any results
print_fault($result); //print any faults

sub print_result {my ($result)=@_;if($result->body&&$result-body->'insertResponse'){
  my %keyHash=$%{ $result->body->'insertResponse' };%foreach $k(keys %keyHash)print "name=$k   value=
  $keyHash{$k}\n";}}

sub print_fault {my ($result)=@_;if($result->fault){print "faultcode=".
  $result->fault->'faultcode' ."\n";print "faultstring=".
  $result->fault->'faultstring' ."\n";print "detail=".
  $result->fault->'detail' ."\n";}}

© 2017 ServiceNow. All rights reserved.
The following is the result printed out by the Perl script on the console

```plaintext
name=display_value   value=INC10011
name=status   value=inserted
name=table   value=incident
name=display_name   value=number
name=sys_id   value=cd45649c0a0b2b006f27649d6bd2c
```

The following image shows the resultant row created for the import set table Notification (imp_notification).

![Image showing the resultant row created for the import set table Notification (imp_notification).](image)

**Figure 448: WS lset Perl**

**Easy import**

Easy import is a simplified import process that removes the need to use a transform map by providing a template that contains only the columns you want to import.

An administrator can use easy import from any list. You can import data to tables within the current scope, and tables that grant write access to other applications.

**Use easy import**

Follow the set workflow to import data using Easy Import.

**Download an import template**

Select the list you want to import data to and create an import template from that list.

Role required: admin

1. Navigate to any list, such as **Self-Service > Incidents**.
2. From the list view, right-click the column headings.
3. Select **Import**.
4. Select the import type.

**Table 368: Import template**

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>Use this template to add new records to a table. The current list filter determines what columns the template contains.</td>
</tr>
<tr>
<td>Update</td>
<td>Use this template to change values within existing records in a table. This template contains one row for each record in the list.</td>
</tr>
</tbody>
</table>

5. Optional: Clear the **Include all fields in the template?** check box to include only columns that appear in the list in the template.

   Leave this check box selected to include all columns from the table in the template, even those columns that are hidden in the list.

6. Click **Create Excel Template**.
ServiceNow creates an import template that contains all fields from the current table, including those fields not displayed in the list.

**Note:** Each template contains a Directions tab describing how to use the template.

Add a record in the template
Add rows to the template to create new records.
Role required: admin

**Note:** The template only imports records on the first page of the template. The import process ignores all other pages.

Figure 449: Easy import insert template

1. Add a row for each record you want to add to the table. Each column corresponds to a field in the target table. Use the UI field hints to learn about the purpose and content of the field.

   The template validates values as you enter them and highlights cells containing invalid data. While you do not have to provide a value for every column, certain fields may fail validation if you do not provide a value.

2. Review any highlighted cell and resolve the issue.

3. When you finish entering new rows, save the spreadsheet as an Excel Workbook (.xls).

   Imports do not currently support .xlsx files.

The insert templates use special processing for certain fields:

- **Created by Field:** The insert template always lists the system administrator user (admin role) in the Created by field regardless of whether the import includes a Created by column. Even if you provide a different user value in the template, the import process overwrites this value with the system administrator user.
• Blank Values: Leaving a cell blank in the insert template results in a blank value in the imported record, provided the field supports a blank value. Fields that do not support a blank value will produce a validation error if you provide a blank value.

Update a record in the template
Modify rows in the template to update existing records.
Role required: admin

**Note:** The template only imports records on the first page of the template. The import process ignores all other pages.

---

**Figure 450: Easy import update template**

1. Edit the row for each existing record to be updated. Replace existing values with new values. Use the UI field hints to learn about the purpose and content of the field.

   The template validates values as you enter them and highlights cells containing invalid data. While you do not have to provide a value for every column, certain fields may fail validation if you do not provide a value.

2. Review any highlighted cell and resolve the issue.

3. When you finish updating existing rows, save the spreadsheet as an Excel Workbook (.xls).

Imports do not currently support .xlsx files.

The insert templates use special processing for certain fields:

- Updated by Field: The update template always lists the system administrator user (admin role) in the Updated by field regardless of whether the import includes an Updated by column. Even if you provide a different user value in the template, the import process overwrites this value with the system administrator user.

- Blank Values: The update template uses these rules when handling blank values in spreadsheet cells.
Import a record from the template

After updating the import template with new data, import the template to your instance.

Role required: admin

1. Navigate to the same list you used to download the import template.
2. From the list view, right-click the column heading and select **Import**.

3. From **Insert or update**, select the import type.
4. From **File**, select the import template to import.
5. Click **Upload and Preview Import Data**.

ServiceNow validates the template.

If the template passes validation and the imported preview matches your expectations, click **Complete Import** to import records into the current table. If the template fails validation, review the errors. Click **Ignore Errors and Complete Import** to continue the import by skipping any bad data cells.

---

**Controlling the easy import template row limit**

You can control the number of rows included in an Easy Import template.

---

<table>
<thead>
<tr>
<th>Original Value</th>
<th>New Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank value</td>
<td>Any non-blank value</td>
<td>Valid update: the update changes the blank value to the new value.</td>
</tr>
<tr>
<td>Any non-blank value</td>
<td>Blank value</td>
<td>Invalid update: the update preserves the existing non-blank value.</td>
</tr>
</tbody>
</table>
By default, all templates generated using Easy Import contain at most 10,000 rows. If a table contains more than 10,000 records, only the first 10,000 are exported.

You can modify this limit by setting the glide.import_template.row_limit property. Add this property to the System Properties [sys_properties] table to specify a different value.

*Easy import template validation*
Each import template must pass two sets of validation.

### Table 370: Template validation

<table>
<thead>
<tr>
<th>Type of validation</th>
<th>Stored in</th>
<th>Results of failing validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client-side validation</td>
<td>Import template</td>
<td>The template highlights the invalid cell.</td>
</tr>
<tr>
<td>Server-side validation</td>
<td>Instance</td>
<td>The import pauses and displays an error message.</td>
</tr>
</tbody>
</table>

**Template data validation**

Some cells in the template contain calculations or validation rules. These rules are intended to help you enter data that is appropriate for the field (column).
If you want to paste data from another spreadsheet, use the Paste Special command to paste only values or match the existing formatting.

Figure 451: Easy import paste special
Template type validations
Server-side validation verifies that the import template contains the necessary format.

Table 371: Template type validation

<table>
<thead>
<tr>
<th>Template type</th>
<th>Validation done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert template</td>
<td>• Template contains a header row with valid field names.</td>
</tr>
<tr>
<td></td>
<td>• Template contains one or more data rows.</td>
</tr>
<tr>
<td></td>
<td>• Template does not contain a sys_id column.</td>
</tr>
</tbody>
</table>
Data type validations
Both client-side and server-side validation verify that template cells contain data that matches the ServiceNow field type.

Table 372: Data type validations

<table>
<thead>
<tr>
<th>ServiceNow field type</th>
<th>Validation done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choice list</td>
<td>Cell contains a value that matches a choice option from the choice list.</td>
</tr>
<tr>
<td>Integer</td>
<td>Cell contains a number without any decimal points.</td>
</tr>
<tr>
<td>Reference</td>
<td>Cell contains a value that matches an existing related record. This validation applies only to fields in which the list of possible choices is relatively small and mostly static. It does not apply to fields such as ‘Assigned To’ in which the list of possible choices is dynamic and large. Validation takes place when the data is inserted into the database, but not in Excel.</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>

Note: When you enter reference field values in the Excel spreadsheet, enter the display value and not the sys_id of the referenced record. Enter this value the same as you would on a form. For example, for a reference to a user record, enter the username.

Troubleshoot import set performance
Review these performance issues to troubleshoot and improve the performance of your import set jobs.

Running business rules during transform
Running business rules during transform may cause the transform to take longer than expected, or cause the instance to slow down.

Becomes an issue: When importing a very large amount of data. For example, importing all data from an old system.

Symptoms: The transform takes much longer than expected. Also, the entire instance may be slow during that time.

How to avoid this: Do not run items like business rules, workflows, approval engines, and so on during a transform unless you want all insert and update business rules, notifications, and workflows to run. For
example, when importing all data from an old system, you may not want notifications to run. To disable these items from running and to cease auditing and field normalization within the transform map for that import, deselect the **Run business rules** check box.

**Figure 453: Transform Map Checkbox**

*Note:* Consider using an **onComplete** transform script to run business logic such as calculations at the end of an import rather than on each record as business rules do.

---

**Slow transform scripts**

Using multiple GlideRecord queries or large loops may slow down transform scripts.

Becomes an issue: When the transform scripts are using multiple GlideRecord queries or looping through large collections of objects for each row. This issue may appear when the transform script is not efficient. In most cases, script goals can be accomplished using built-in functionality within the Import Set application. For example, you can script **case-sensitive coalesce** instead of writing scripts that use GlideRecord queries. GlideRecord queries typically slow down the import.

Symptoms: The transform takes much longer than would be expected. Depending on the script, the entire instance may be slow during that time.

How to avoid this: Use base system functionality whenever possible instead of writing custom scripts and if you do write scripts, avoid writing complicated scripts that use GlideRecord queries.

**Importing data that has not changed**

Repeatedly importing data that has not changed leads to many skipped rows.

Becomes an issue: When a customer is importing data from a table that is very large and most of the records are not getting updated on a regular basis.

Symptoms: The import set takes longer than expected. Under **System Import Sets > Progress**, expect to see an import with a Total count that is very high with a Skipped count that is also very high - this is found under the **Message** column. Indicating that most of the records imported had not actually changed. These records did not need to be imported.

How to avoid this: If you are running a JDBC import, use the last run datetime option in your import set **DataSource**. For a type of File import, be sure that whatever is generating your files is only adding data that is new, or has been changed.

**Coalescing on non-indexed fields**

Coalescing on non-indexed fields with a large amount of data may cause transforms to slow down.
Becomes an issue: When matching on fields that are not indexed, this causes the transform stage of an import to run slowly. However, it only becomes an issue if there is a large enough amount of data. In extreme cases, this causes performance issues with the database due to added load.

Symptoms: Time spent in the transform stage of the import is large relative to the time taken to load the data. Expect to see high transform times.

How to avoid this: If possible, you should coalesce on a field that unique and already indexed. To determine if a field is already indexed, navigate to System Definition > Tables & Columns and find the table. In the list of columns for that table, an indexed column has a blue icon with an i next to it if indexed. For assistance indexing a field contact ServiceNow Technical Support.

Running imports simultaneously
Running imports simultaneously may cause excessive load on the database.

Becomes an issue: When importing large amounts of data puts an additional load on the database. For example, importing 500,000 users and importing 200,000 configuration items at the same time. This can have a significant performance impact on all queries on the system due to the increased load on the database. This issue is especially severe when two imports are importing to the same table. In such a case, there is a possible contention issue for the table. Additionally, depending on which table is involved in processing, this can severely degrade performance of the import and the instance.

Symptoms: Multiple simultaneous imports running slowly combined with load on the database. You see large numbers of inserts and updates along; and if there is enough load or contention, high IO Wait times.

How to avoid this: Stagger your imports so they do not overlap.

Large import set tables
Failing to clean import set tables may lead to those tables becoming cluttered and slow.

Becomes an issue: When the Import Set Deleter job is not running.

Symptoms: This is a size issue. If the import sets are not cleaned on a regular basis (a cleanup is recommended after seven days worth of data) the table fills causing imports to stop.

How to avoid this: Verify the Import Set Deleter job is running. If it is not currently running, contact customer support as they will truncate all import set tables before enabling this job.

Altering table schema during import
Changing the table schema, such as by importing a new column locks the import set table.

Becomes an issue: Any time a new column is imported, the entire import set table is locked during that schema change and depending on the size of the table, can take between five and ten minutes. During that time, no data can be selected or inserted. If that table is not used often, this may not cause any problems. However, if that table is frequently used, for example the LDAP import table, issues may arise.

Symptoms: The symptoms of this problem may vary. In our example of the LDAP import table, any transactions requiring a query of the LDAP import table will have to wait until the schema change has completed. As the LDAP import table is used when logging into the instance, no users are able to log in while the schema change is happening.

How to avoid this: Truncate the import table before importing with a new column.

Importing very large data sets
Importing a very large data set takes longer than importing multiple smaller data sets.

Becomes an issue: When very large data sets are imported in a single job.

Symptoms: The import job takes a long time to complete.

How to avoid this: Break a very large data set into multiple, smaller jobs for faster results. Consider import sets under 100,000 records as a guideline. For example, importing 10 sets of 100,000 records completes faster than one import of 1 million records even though the total data imported is the same.
Importing from another ServiceNow instance

There are several methods for moving data from one instance to another. All of these methods assume you need to periodically move a collection of records from a table on one instance to a table on another instance. If you want an action on an individual record to trigger a corresponding action on another instance, consider using a REST web service instead.

Identifying Needed Related Records

All import processes work by fetching data from one table at a time. This process can produce unexpected results if the incoming data references new records in other tables such as users, locations, or companies. To prevent importing broken references to other tables you must identify the connections between your tables and import the common data prior to importing the application data.

For example, prior to importing incident data, you might want to first import related configuration items, users, and locations. Furthermore you might want to import incidents before importing problems or changes that refer to these incident records.

Available Import Processes

The system offers the following processes to move data from one instance to another.

- Export and import XML records directly from lists
- Create XML import sets and transform data as needed
- Export and import CSV files from lists

XML records from lists
You can directly export records as XML from any list or form.

This export process preserves all record field values including system generated field values such as:

- Sys ID
- Creation date
- Update date

The XML import process directly inserts records into the target table. During the import the system:

- Ignores any business rules that normally apply to the table.
- Does not provide any opportunity to transform incoming data.
- Automatically matches a reference field's display value to the local Sys ID for some tables.

This process is good for directly copying records from one system to another but does not remove the need to know the relationships between tables. If you need validate or transform data or reconcile the sys ID values of reference fields, use an XML import set instead.

Automatic matching of display values

During the import of XML records, the system attempts to match some reference field display values to a local sys_id value.

If the system finds an existing record with a matching display value on the local instance, the import uses the sys_id of the existing record rather than the sys_id of the imported record.

For example, suppose you export an incident record that is assigned to the user John Smith. In the exported XML file there is an entry such as:

```xml
<intcident>
```
This user already exists on the target instance but has a different sys_id value such as:

```xml
<sys_user>
  <name>John Smith</name>
  <sys_id>18cab8de2be80200c5244f74b4da15f7</sys_id>
</sys_user>
```

Since the display value matches an existing record, the system uses the local instance's existing sys_id value for the reference field such as:

```xml
<incident>
  <assigned_to display_value="John Smith">18cab8de2be80200c5244f74b4da15f7</assigned_to>
</incident>
```

The system can match display values for the following tables.

- User [sys_user]
- Group [sys_user_group]
- Role [sys_user_role]
- Group Roles [sys_group_has_role]

**Using XML import sets**

Use an XML import set to import data from another instance.

This method allows you to:

- Apply business rules
- Transform incoming data
- Reconcile sys_id values if necessary

**Note:** Administrators cannot set the update_synch attribute in Dictionary records to move data as part of update sets. This attribute was being used incorrectly to migrate data in large tables and was causing significant performance problems.

Create an XML data source

Data sources are used to create an import set so that data can be processed, if necessary, prior to being mapped onto a production table.

**Note:** To import using XML with High Security Settings enabled, you must possess elevated privileges.

1. Navigate to **System Import Sets > Data Sources**.
2. Click **New**.
3. Complete the form using the following values:
   - **Format:** XML
   - **Import set table label:** `<import_set_table_label>`
   - **Xpath root node:** `//<source_table_name>`
• **Expand node children**: true.
• **File retrieval method**: HTTPS
• **File path**: `<source_table_name>.do?XML`
• **Server**: `<instance name>.service-now.com`
• **User name**: A user account on the remote instance. The username used cannot be an email address.
• **Password**: The password for the same remote account.

4. Click **Submit**.
5. Click **Test Load 20 Records** to create the import set table and to ensure your data source is functional.

---

**Note:**
The 20 loaded records cannot be transformed and are for testing purposes only.

---

Create a transform map

A transform map is a set of field maps that determine the relationships between fields in an import set and fields in an existing ServiceNow table, such as Incidents [incident] or Users [sys_user].

After creating a transform map, you can reuse it to map data from another import set to the same ServiceNow table.

1. Open the data source record that you just created.
2. In the **Transforms** related list, click **New**.
3. Populate all the usual fields and be sure to choose the proper **Target table**.
4. Submit the record.
5. Click **Auto map matching fields**.

Add a field map for `sys_id`

1. Open the table transform map record you just created.
2. In the **Field Map** related list, click **New**.
3. Complete the form, using the following values
   - **Source field**: `sys_id`
   - **Target field**: `Sys ID`
   - **Coalesce**: Select the check box.
4. Click **Submit**.

Add onBefore scripts to the transform map

Create one onBefore transform map script to copy the `sys_id` of new records from the source to the target instance. Create a second onBefore transform map script to identify records on the target instance that have the same unique values but different `sys_id` values.

1. Open the table transform map record you just created.
2. In the **Transform Scripts** related list, click **New**.
3. In the **When** field, select onBefore.
4. Enter the following Script:

   ```javascript
   if (action == "insert") { target. setNewGuid (source. u_sys_id ); }
   ```

5. Click **Submit**.
6. In the **Transform Scripts** related list, click **New**.
7. In the **When** field, select **onBefore**.
8. Enter the following **Script**:

```javascript
/**
 * This script queries for a uniquely identifying value of the referenced record and then
 * updates the target reference field with the sys_id of the matching target record.
 * This sample assumes:
 * 1) The target table contains an assigned_to field which is a reference field.
 * 2) The reference field references the User [sys_user] table.
 * 3) You can use the email field to uniquely identify users.
 *     Alternatively you could use the user_name field.
 */
var ref = new GlideRecord("sys_user"); //Replace sys_user with any reference table
ref.addQuery("email", source.email); //Replace email with any unique field
ref.query();
if(ref.next()){
    target.assigned_to = ref.sys_id; //Replace assigned_to with any reference field
}
```

9. Click **Submit**.

**Retrieving data from a CSV formatted file**

In this method, you import data from another ServiceNow instance using an HTTPS data source to return a CSV formatted file containing the rows to be imported.

This approach uses the Import Sets application to retrieve the data from the source and import it to the destination.
1. Create a new data source by navigating to **Import Sets > Data Sources** on the destination instance.
2. Provide the following field values:
   - **Import set table name**: Select a table.
   - **Type**: File
   - **Format**: CSV
   - **File retrieval method**: HTTPS

   ![Data Source](image)

   **Figure 454: HTTPS Data Sources**

   - **File path**: `/incident\do?CSV`
   - **SCP authentication method**: `-- None --`
   - **Server**: `test.service-now.com`
   - **Private keyfile**

**Note:**

If the **Server**, **File path**, **Username**, and **Password** fields are not visible when you select HTTPS, change the file retrieval method temporarily to SCP and enter this information. Remember to reset the **File retrieval method** to HTTPS after you enter these fields.
• **File path:** `incident.do?CSV`

  **Note:** To import specific change records, you can add qualifiers to the path. For example, the following path would return all active change records: `incident.do ?CSV &sysparm_query =active = true "`

• **User name** and **Password:** Enter the user name and password for a valid user on the destination instance.

3. Click the **Test load 20 records** related link to verify that the import is configured correctly.

4. After setting up the data source, configure a standard transform map.

### Export sets

Export sets allow you to push data from an instance to an external file.

You can export data to a remote system using a MID server.

**Note:** You must use a MID server from the Geneva release. Earlier MID server releases do not support export sets.

### Create an export set

Create an export set to export records from your instance to a file on a MID Server.

**Role required:** export_set_admin

**Note:** Export sets do not export attachments to records. To download an attachment, either use the REST Attachment API (HTTP request originates from a third-party HTTP client), or use the outbound REST Message module to send the attachment from the instance (HTTP request originates from the instance).

1. Navigate to **System Export Sets > Create Export Set.**
2. Enter a descriptive **Name** for the export set.
3. In the **What to export** section, define what data to export in one of these ways.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Yes and select an Export Definition record.</td>
<td>Use this configuration if you have already created an export definition record specifying what data to export.</td>
</tr>
<tr>
<td><strong>Select No and select a table to export data from.</strong></td>
<td>Use this configuration if you have not created an export definition record. A new export definition record is created automatically using the selected table that includes fields from the default list view for that table. You can modify the export definition record as needed after creating the export set.</td>
</tr>
</tbody>
</table>

4. In the **Where to export to** section, define where you want to export data to in one of these ways.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Yes and select an Export Target record.</td>
<td>Use this configuration if you have already created an export target record specifying where to export data to.</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
Select No and select a MID Server, and specify a path on the MID Server to save the exported data to. | Use this configuration if you have not created an export target record. A new export target record is created automatically for the selected MID Server and file path. You can modify the export target record as needed after creating the export set.

5. Click **Submit**.

After creating the export set, the Export Set form appears. You can configure advanced options from the form such as specifying a data format or scheduling recurring exports.

**Export set fields**

These fields appear on the Export Set form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for this export set.</td>
</tr>
<tr>
<td>File name</td>
<td>Enter a name for the target file.</td>
</tr>
<tr>
<td>Format</td>
<td>Select the format of the target file, such as CSV.</td>
</tr>
<tr>
<td>Append timestamp</td>
<td>Select this check box to append the current date and time to the name of the exported file.</td>
</tr>
<tr>
<td>Export definition</td>
<td>Select the export definition that specifies the data to export.</td>
</tr>
<tr>
<td>Export target</td>
<td>Select the export target that specifies the location you want to export to.</td>
</tr>
</tbody>
</table>

Create an export set from a list

You can quickly create an export set from a list.

Role required: export_set_admin

1. Navigate to a list of records.
2. Right-click on the list header and select **Export** > **Export Set**. The Export Set form appears. An export definition is created automatically based on the list.
3. Configure the export set as needed.
4. Click **Submit**.

**Export definition**

An export definition specifies the data to be exported in an export set.

An export definition specifies a table, one or more fields, and optionally a filter to limit the included records.

Create an export definition

Create an export definition to define what data to export in an export set.

Role required: export_set_admin

1. Navigate to **System Export Sets** > **Export Definition**.
2. Click **New**.
3. Enter a descriptive Name for the export definition.
4. Select the Table to export data from.
5. Select one or more Fields from the selected table to export data from.
6. Optional: Specify a Filter to export only certain records from the selected table.
   Specifying a filter condition on the Created (sys_created_on) or Updated (sys_updated_on) fields may prevent scheduled data exports from using delta exports functionality. Do not specify filter conditions on these fields if you intend to use scheduled delta exports.

Export target

An export target defines the destination file an export set saves data to.

You must use a MID server as the export destination.

Create an export target

Define where to export data to by creating an export target.

Role required: export_set_admin

Navigate to System Export Sets > Export Targets and create a new record (see table for field descriptions).

Table 374: Export target fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for the export target.</td>
</tr>
<tr>
<td>MID Server</td>
<td>Select the MID Server to export to. The MID Server must have a Validated value of Yes and a Status value of Up.</td>
</tr>
<tr>
<td>Note:</td>
<td>You must use a MID server from the Geneva release. Earlier MID server releases do not support export sets.</td>
</tr>
<tr>
<td>File Path</td>
<td>Specify a path relative to the MID Server directory to save the exported data to. The user account used to run the MID Server must have permission to write to this location.</td>
</tr>
</tbody>
</table>

Export set supported file types

Export sets support multiple file types for exporting and pushing data.

File types

You can export to these formats:

- CSV
- XML
- Excel
Scheduled export sets

You can schedule an export set to regularly export data.

By setting up a recurring, scheduled export you are able to regularly push data to an external location, such as for reporting or for preserving a snapshot of the data.

You can configure scheduled export sets to export only new or changed records using delta exports.

Schedule an export
You can schedule an export to regularly push data from an export set to a remote destination.

Role required: export_set_scheduler or export_set_admin

1. Navigate to System Export Sets > Scheduled Exports.
2. Click New.
3. Fill in the following fields.

Table 375: Schedule Data Export fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for this scheduled export.</td>
</tr>
<tr>
<td>Export set</td>
<td>Select the export set to schedule.</td>
</tr>
<tr>
<td>Run as</td>
<td>Select the user to run the scheduled export as. Ensure the user has any roles needed to view the data to export.</td>
</tr>
<tr>
<td>Run</td>
<td>Select the frequency for scheduled export.</td>
</tr>
<tr>
<td>Day</td>
<td>Select which day of the week or month to run this scheduled export on. This field appears when the Run field is set to Weekly or Monthly</td>
</tr>
<tr>
<td>Repeat interval</td>
<td>Enter the number of days and hours to wait before repeating this scheduled export. This field appears when the Run field is set to Periodically.</td>
</tr>
<tr>
<td>Time</td>
<td>Enter the time of day to run the scheduled export.</td>
</tr>
<tr>
<td>Delta Exports</td>
<td>Select this check box to enable delta functionality for this scheduled export. When selected, only new or changed records are exported.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Delta on or after</td>
<td>Select <strong>Updated</strong> to export all records that changed since the last export, including new records. Select <strong>Created</strong> to export only new records created after the last export.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the export definition already includes a filter condition on the <code>sys_updated_on</code> or <code>sys_created_on</code> fields, enabling delta exports will prevent you from saving the scheduled data export record. An error appears at the top of the form in this event.</td>
</tr>
<tr>
<td>Last export scheduled run</td>
<td>Leave this field blank to export all records from the export definition. When using delta exports, this field is set automatically each time the scheduled export runs. Records updated or created after this date and time are included in the export delta, depending on the <strong>Delta on or after</strong> value.</td>
</tr>
</tbody>
</table>

### Scripts

<table>
<thead>
<tr>
<th>Conditional</th>
<th>Select this check box to run the schedule only when a condition is met.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>Write the script to be used to evaluate whether the export should run. This field is visible if you select the <strong>Conditional</strong> check box.</td>
</tr>
<tr>
<td>Execute pre-export script</td>
<td>Select this check box to run a script before the export begins.</td>
</tr>
<tr>
<td>Pre script</td>
<td>Write the script you want to run before the data is exported. This field is visible if you selected the <strong>Execute pre-export script</strong> check box.</td>
</tr>
<tr>
<td>Execute post-export script</td>
<td>Select this check box to run a script after the export finishes.</td>
</tr>
<tr>
<td>Post script</td>
<td>Write the script you want to run after the export finishes. This field is visible if you selected the <strong>Execute post-export script</strong> check box.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

You can click the **Execute Now** button to run the export immediately and confirm the data is exported as expected.

**Scheduled export set scripting options**

When scheduling an export set, you can access certain JavaScript objects.

Use these objects in scheduled export set scripts, such as the **Pre-script** and **Post-script** fields.
Table 376: Export set scripting objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancel</td>
<td>Set this object to true to stop the export action. Any child export sets are also cancelled if the parent is cancelled. This object is available only in the Pre-script field.</td>
<td>Use the Pre-script field to evaluate the conditions of the export and determine whether to cancel the export process. To cancel the import process, use the following call:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cancel = true;</td>
</tr>
<tr>
<td>export_set</td>
<td>Get the GlideRecord object for the new export set. This variable allows you to query the following columns from the sys_export_set table: name, sys_id, state, table_name. This object is available in both the Pre-script and Post-script fields.</td>
<td>If you want to use information from the export set, you can specify one of the properties of the export_set variable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>var x = export_set.sys_id;</td>
</tr>
</tbody>
</table>

Export set history

An export set history record is automatically created whenever an export set runs, either scheduled or manually started by a user.

Export set history records allow you to monitor and troubleshoot the progress of export sets.

Fields on the Export History form display the current status of the export set. The Export Log related list shows a detailed list of changes to the export set status.

Cancel an export set

You can cancel a running export set to stop the data from being sent to a MID server.

Role required: export_set_admin

You can cancel only exports that have not yet been sent to a MID server. You cannot cancel an export after it reaches the MID Server Processing state.

1. Navigate to System Export Sets > Export Sets.
2. Select an export set.
3. In the Export Histories related list, select a record with the State value of Exporting.
4. Click Cancel Export Set.

Export set history fields

These fields appear on the Export History form.
### Table 377: Export History fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export set</td>
<td>The export set that was run.</td>
</tr>
<tr>
<td>Started</td>
<td>The date and time the export started.</td>
</tr>
<tr>
<td>Completed</td>
<td>The date and time the export completed.</td>
</tr>
<tr>
<td>Export time</td>
<td>The amount of time it took to export the data, excluding the time it took to transfer the data to the target.</td>
</tr>
<tr>
<td>Run time</td>
<td>The amount of time it took to export the data, including the time it took to transfer the data to the target.</td>
</tr>
<tr>
<td>State</td>
<td>The current state of the export.</td>
</tr>
<tr>
<td>Delta export</td>
<td>The export used delta export functionality if this check box is selected.</td>
</tr>
<tr>
<td>ECC queue</td>
<td>The ECC Queue record associated with this export set run. This field is visible only to users with the admin role.</td>
</tr>
<tr>
<td>MID Server Attachment</td>
<td>A MID Server attachment record containing a copy of the exported file. The exported file is attached to this record in addition to being saved on the MID Server.</td>
</tr>
</tbody>
</table>

The **State** field may have the following values.

### Table 378: States

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exporting</td>
<td>The export set is creating a file with the data to export.</td>
</tr>
<tr>
<td>MID Server Processing</td>
<td>The export set successfully created a file with the data to export. The file is being processed by the MID Server.</td>
</tr>
<tr>
<td>Completed</td>
<td>The export set successfully ran and sent the exported file to the target.</td>
</tr>
<tr>
<td>Completed with errors</td>
<td>The export set ran, but encountered one or more errors.</td>
</tr>
<tr>
<td>Cancel requested</td>
<td>A user requested that the export set be cancelled.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>The export set was cancelled.</td>
</tr>
</tbody>
</table>

### Export Set properties

These properties control the behavior of export sets.
### Table 379: Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.scheduled_export.stop_on_parent_error</td>
<td>When true, if a parent scheduled export set encounters an error, scheduled child export sets do not run. This property is false by default.</td>
</tr>
</tbody>
</table>

### Table administration

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table.

Applications use tables and records to manage data and processes, such as Incident, Problem, and CMDB. Tables can extend other tables, creating parent tables and child tables.

Administrators can use these tools for viewing and modifying the database structure:

- Tables module: provides a list of all tables in the database.
- Tables & Columns module: provides a list of all existing tables, with columns, column attributes, and indexes.
- Schema map: provides a graphical representation of the relationships between tables.
- Data dictionary tables: contain additional information that defines database elements.

### Create a table

Administrators and application developers can create custom tables to store application data.

Role required: admin

1. Navigate to **System Definition > Tables**.
2. Click **New** and complete the form.

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Enter a unique label for the table (such as <strong>Laptops</strong> or <strong>Thin Clients</strong>). The label appears on list and form views for the table. Updating the <strong>Label</strong> field also updates the label record in the language file for the current language.</td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>Edit the table name, which is automatically populated based on the table label and a prefix as follows:</td>
</tr>
<tr>
<td></td>
<td>• For a table in a scoped application, the name is prefixed with a namespace identifier to indicate that it is part of an application.</td>
</tr>
<tr>
<td></td>
<td>Preface the name of the table with u_cmdb_ci_ to make it similar to the other CMDB classes (for example, u_cmdb_ci_laptop).</td>
</tr>
<tr>
<td></td>
<td>You cannot modify the prefix; however, you can modify the rest of the table name. The name can contain only lowercase, alphanumeric ASCII characters and underscores (_).</td>
</tr>
<tr>
<td>Extends Table</td>
<td>Select the table to extend. Extending a base table incorporates all of the fields of the original table and creates system fields for the new table. You can extend tables that are marked as extensible if they are in the same scope or if they allow configuration from other scopes.</td>
</tr>
<tr>
<td></td>
<td>For example, if the new class is Laptops, which is a subclass of Computers, select the cmdb_ci_computer table. If the new class is a top-level class, select the cmdb_ci table.</td>
</tr>
<tr>
<td></td>
<td>This option is available only when creating a table.</td>
</tr>
<tr>
<td>Application</td>
<td>[Read only] Displays the application associated with this table. If you are working on an application or are creating a table from an application record, the field defaults to that application. Otherwise, the field defaults to Global. Any records that are created from the table record, such as modules and security rules, are assigned to this application by default.</td>
</tr>
<tr>
<td>Create module</td>
<td>Select the check box and then complete the <strong>Add module to menu</strong> field to create a list module in the application menu.</td>
</tr>
<tr>
<td></td>
<td>This option is available only when creating a table.</td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Add module to menu</td>
<td>Select an existing menu or select <strong>Create new</strong> and enter a new menu name. This option is available only when the <strong>Create module</strong> check box is selected.</td>
</tr>
</tbody>
</table>

3. **In** the **Columns** section, use the **Table Columns** embedded list to add columns to the table.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column label</td>
<td>Define a unique label for the column. The label appears on list headers and form fields for the column.</td>
</tr>
<tr>
<td></td>
<td>• When you update the <strong>Column label</strong> field, the system also updates the label in the language file for the current language.</td>
</tr>
<tr>
<td></td>
<td>• When you create a new column, the column name is automatically populated based on the label. The name is automatically prefixed with <code>u_</code> to indicate that it is custom. For example, if you enter <strong>Activity Description</strong> as the column label, the column name defaults to <code>u_activity_description</code>.</td>
</tr>
<tr>
<td>Type</td>
<td>[Mandatory] Define the field type for the column. To preserve existing data, only change fields between the same basic type (for example, <strong>Choice</strong> and <strong>String</strong>). A warning appears if a change to a custom field will result in data loss. For a base system field, you cannot make a change that will result in data loss.</td>
</tr>
<tr>
<td>Reference</td>
<td>Make the field into a reference field by entering the referenced table name.</td>
</tr>
</tbody>
</table>

**Note:** Dynamic reference creation is enabled for this field. So, if you enter a table name that does not match an existing table, a new table is created when you save changes to the current table record. If the current table has a module in the application navigator, then a module for the newly created table is automatically created in the same application menu.
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>[String fields only] Limit the length of the field. A length of under 254 appears as a single-line text field. Anything 255 characters or over appears as a multi-line text box.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• You can only change this value if the <strong>Type</strong> of the field is <strong>String</strong>. Changes for any other type of field are ignored.</td>
</tr>
<tr>
<td></td>
<td>• Users on an Oracle instance cannot increase the maximum length of a string field to anything greater than 4000 through the application UI because this requires the CLOB datatype in Oracle. To increase beyond this size, log an incident with ServiceNow Technical Support to request the change.</td>
</tr>
<tr>
<td></td>
<td>• To prevent data from being lost, only decrease the length of a string field when you are developing a new application and not when a field contains data. A warning appears if a change to a custom field will result in data loss. For a base system field, you cannot make a change that will result in data loss.</td>
</tr>
<tr>
<td>Default value</td>
<td>Specify the default value of the field for any new record. Ensure that this value uses the correct field type. For example, an integer field can use a default value of 2 but cannot use a default value of two. These values can be overridden with dictionary overrides.</td>
</tr>
<tr>
<td>Display</td>
<td>Indicate whether this field is the display value for reference fields (appears on records that reference this table).</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This option does not control whether or not this field is displayed on lists or forms.</td>
</tr>
</tbody>
</table>

4. In the **Controls** section, define additional table options.
<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensible</td>
<td>Select the check box to allow other tables to extend this table. Clear the check box to prevent the creation of additional child tables; existing child tables remain unchanged.</td>
</tr>
<tr>
<td>Live feed</td>
<td>Select the check box to enable record feeds for the table. This option adds the <strong>Show Live Feed</strong> icon (ês) in the form header.</td>
</tr>
<tr>
<td>Auto-number</td>
<td>Select the check box, and then define the number format to add an auto-numbered field to the table. The check box is available only when a number format does not exist for the table. Otherwise, you can edit the existing number format.</td>
</tr>
<tr>
<td>Create access controls</td>
<td>Select the check box and then complete the <strong>User role</strong> field to create basic security rules for the table.</td>
</tr>
<tr>
<td>User role</td>
<td>Enter a new name or select an existing user role that is required to access this table. This option is available only when the <strong>Create access controls</strong> check box is selected.</td>
</tr>
</tbody>
</table>

5. In the **Application Access** section, define the scope protection for the table.

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can read</td>
<td>Select the check box to allow script objects from other application scopes to read records stored in this table. This option offers runtime protection. For example, a script in another application can query data on this table. You must first select read access to grant any other API record operation.</td>
</tr>
<tr>
<td>Can create</td>
<td>Select the check box to allow script objects from other application scopes to create records in this table. This option offers runtime protection. For example, a script in another application can insert a new record in this table. This option is available only when the <strong>Can read</strong> check box is selected. Clear the check box to prevent script objects from other application scopes from creating records in this table.</td>
</tr>
</tbody>
</table>
### Control

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can update</td>
</tr>
<tr>
<td>Can delete</td>
</tr>
<tr>
<td>Allow access to this table via web services</td>
</tr>
<tr>
<td>Allow configuration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can update</td>
<td>Select the check box to allow script objects from other application scopes to modify records stored in this table. This option offers runtime protection. For example, a script in another application can modify a field value on this table. This option is available only when the Can read check box is selected. Clear the check box to prevent script objects from other application scopes from modifying data stored in this table.</td>
</tr>
<tr>
<td>Can delete</td>
<td>Select the check box to allow script objects from other application scopes to delete records from this table. This option offers runtime protection. For example, a script in another application can remove a record from this table. This option is available only when the Can read check box is selected. Clear the check box to prevent script objects from other application scopes from deleting records from this table.</td>
</tr>
<tr>
<td>Allow access to this table via web services</td>
<td>Select the check box to allow users to make inbound web service queries to this table. This option offers both design-time and runtime protection. The user performing the query must have the correct permissions to access this table, even when this check box is selected. Clear the check box to prevent users from making web service queries to this table.</td>
</tr>
<tr>
<td>Allow configuration</td>
<td>Select the check box to allow applications from other application scopes to create configuration records for this table that change its functionality. For example, an application designer can select this table from the Tables list on business rules, client scripts, or UI actions. This option offers design-time protection. Clear the check box to prevent application designers from selecting this table when creating configuration records.</td>
</tr>
</tbody>
</table>

6. Click Submit.
Global default fields

When you create a new custom table, several fields appear in the **Table Columns** embedded list. For all tables, required system fields are added automatically. You cannot delete or modify these fields.

For tables that extend another table, fields on the parent table also appear on the **Table Columns** embedded list for the current table. If you modify these fields, remember that all changes to fields on the parent table also affect all child tables, not just the current table.

These required system fields are added to all tables:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class [sys_class_name]</td>
<td>System Class Name</td>
<td>If the table is extensible, a string field that indicates which child table contains the record.</td>
</tr>
<tr>
<td>Created [sys_created_on]</td>
<td>Date/Time</td>
<td>A time-stamp field that indicates when a record was created.</td>
</tr>
<tr>
<td>Created by [sys_created_by]</td>
<td>String</td>
<td>A string field that indicates the user who created the record.</td>
</tr>
<tr>
<td>Sys_id [sys_id]</td>
<td>Sys ID</td>
<td>The unique record identifier for the record.</td>
</tr>
<tr>
<td>Updates [sys_mod_count]</td>
<td>Integer</td>
<td>A numeric field that counts the number of updates for this record since record creation.</td>
</tr>
<tr>
<td>Updated by [sys_updated_by]</td>
<td>String</td>
<td>A string field that indicates the user who most recently updated the record.</td>
</tr>
<tr>
<td>Updated [sys_updated]</td>
<td>Date/Time</td>
<td>A time-stamp field that indicates the date and time of the most recent update.</td>
</tr>
</tbody>
</table>

Activate restore deleted records

Administrators can activate the Restore Deleted Records plugin to enable this functionality.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.
Create a many-to-many relationship

Many-to-many relationships allow a list to point to a list of entries, rather than to single field. For example, if a knowledge base article points to a list of related configuration items, it uses a related list. Not all lists can be used as related lists, however. For a list to be related to another list, a many-to-many relationship that refers to both tables must exist.

The Many to Many Definitions [sys_m2m] table allows administrators to create custom many-to-many relationships.

1. In the navigation filter, enter sys_m2m.list.
2. Click New.
3. In the From table field, specify a parent table.
4. In the To table field, specify a child table.
   The Many-to-Many form automatically populates the other fields with suggested values.
5. Optional: Edit other field values, if appropriate. Note that Many-to-Many table names cannot exceed 30 characters.

Reference default many-to-many relationships

Some many-to-many relationships are defined by default.

To reference many-to-many relationships that are available in the base system, administrators can enter sys_collection.list in the navigation filter.

---

**Note:** Only use this table to view many-to-many relationships in the base system. To create a new relationship, always use the Many-to-Many Definitions table.

Delete all records from a table

You may decide to delete all the records on a table without deleting the table itself. For example, the administrator may want to delete all incidents on a test instance without deleting the incident table itself.

When you prepare to delete many records from a table, consider the following guidelines to minimize impact on performance.

- Limit the number of records to be deleted in a single delete action to prevent the table from being locked. Use the setLimit() method described at setLimit.
- Minimize triggering an excessive number of business rules as a result of this deletion. Use the setWorkflow(Boolean e) method described at setWorkflow.

---

**Note:** Deleting all records for a table also deletes records from tables that extend the table.

There are several methods for deleting table records. Depending on the number of records to be deleted, choose the method that would be most efficient in your environment. Use these methods with caution. Before performing this procedure, be sure that you do not need any of the records.

- From the Tables and Columns module, complete the following steps.
  a) Navigate to System Definition > Tables and Columns.
  b) Select the table for which to delete records.
Notes: Some system tables do not allow this method of deleting records, for example, you cannot delete all user records [sys_user]. The list of tables does not include system tables that you cannot delete records from.

c) Click **Delete all records**.
d) In the confirmation dialog box, enter delete and click OK.

• From a list view, complete the following steps.
  a) Navigate to the list view that displays the table records, click the context menu, and then navigate to **Show**.
  b) Set the number of rows per page to display the max number.
  c) Select all the rows on the page.
  d) Click **Actions on selected rows**, and then click **Delete**.
  e) Continue deleting all rows on a page until all records are deleted.

The selected table is empty of records. The table still exists, and any references to the table on other tables (such as business rules or reference fields) are preserved.

**Delete a table**

Administrators can delete custom tables that are no longer needed. For example, delete a table from an application that is under development because the business requirements change.

A table is custom if it was created by an administrator and is not part of a system upgrade or plugin activation. Custom table names always begin with u_.

Base system tables cannot be deleted. Additionally, any missing base system tables are recreated when the instance is upgraded. You cannot delete a table if another table extends it.

**Warning:** You must delete the records in the table before you delete the table. If you do not delete the records first, errors can result on the parent table if it references the records. Deleting the records removes any references from the parent table.

**Delete a custom table**

1. Navigate to **System Definition > Tables**.
2. Open the table to delete.
3. Click **Delete All Records**.
   - Deleting all records before deleting the table ensures that the business logic is properly executed (for example, reference cascade rules or other delete business rules). If you do not delete all records from the table first, then you must manually fix any other records or tables that are impacted by the table deletion.
4. Click **Delete**.
5. In the confirmation dialog box, enter delete and click **OK**.

   The table and all items that reference the table are deleted, including:
   - Choice list items
   - Forms, form sections, lists, and related lists
   - Reports and gauges
   - Reference fields that reference the table
   - Access controls

**Restore deleted records**

In some cases, administrators can restore deleted records and references to those records.
In some cases, administrators can restore deleted records and references to those records. For example, if a user inadvertently deletes a user record that was referenced in the **Caller ID** field on several incident records, you may be able to restore the user record as well as the incident field values. You may also be able to restore records that were deleted as a result of a cascade delete action.

There are several methods for restoring deleted records:

- Restore data records without references on tables that audit deletions.
- Restore data records and references on tables that audit deletions (requires the Restore Deleted Records plugin).
- Restore configuration records with the app creator.

**Restore a deleted record and reference**

ServiceNow can track deletions on any table, and references on audited tables, except in these circumstances:

- Record deletions are not tracked on tables with the `no_audit_delete=true` dictionary attribute.
- **Attachments** are not restored when the record is restored.
- References to a restored record are restored only if the reference field is on an audited table.
- References to images using an **Image** type field are not restored.

To restore deleted records and references:

1. Navigate to **System Definition > Deleted Records**.
2. Open a deleted record that you want to restore.
   
   **Note:** You can only restore one deleted record and its associated references at a time.

3. In the Audit Deleted Record form, select **Restore Record and References** under **Related Links**.
   
   **Note:** To restore the record without restoring the references, select **Undelete Record**

   An instructions page appears, explaining the process and its limitations.

4. Click **Restore deleted record**.
   
   A Progress page shows the progress of the restore process, and displays a counter of the references restored.

When the process completes, a Restore Summary itemizes the changes and provides links for viewing the restored record or returning to the deleted records list.

**Restore data records with deletion audits**

You can restore deleted data records on tables that audit deletions.

Limitations for restoring data records include:

- Record deletions are not tracked for tables with the `no_audit_delete=true` dictionary attribute.
- **Attachments** are not restored when the record is restored.
- Deletions from tables with a sys prefix are not audited by default. You can use the app creator to **restore configuration records**, such as Business Rules [sys_script]. You can also **configure specific system tables to audit deletions**.
- References are restored only if the reference field is on an audited table and the Restore Deleted Records plugin is activated.
- References that use an **Image** field type are not restored.
Restore a system table

ServiceNow does not automatically track deletions from tables with a `sys` prefix. You can configure this behavior using these steps.

ServiceNow does not automatically track deletions from tables with a `sys` prefix. To track deletions from these tables, add the table name to the `glide.ui.audit_deleted_tables` property. *Activate the Restore Deleted Records plugin* adds several default values to this property.

To configure which tables are tracked:

1. Navigate to **System Properties > UI Properties**.
2. Locate the **List of system tables (beginning with "sys_", comma separated) that will have the delete audited property**.
3. Add or remove table names. Table names should be separated by commas, without any spaces.
4. Click **Save**.

*Note:* For more information about auditing, see *Understanding the sys audit Table*.

Schema map for tables

The schema map displays the details of tables and their relationships in a visual manner, allowing administrators to view and easily access different parts of the database schema.

The schema map can also be printed directly from a browser.

Schema relationship types supported, and the colors used for them, are:

<table>
<thead>
<tr>
<th>Relationship type</th>
<th>Color used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referenced by</td>
<td>Red</td>
</tr>
<tr>
<td>Referencing</td>
<td>Orange</td>
</tr>
<tr>
<td>Extended by</td>
<td>Green</td>
</tr>
<tr>
<td>Extending</td>
<td>Blue</td>
</tr>
</tbody>
</table>

By default, all these types of relationship are displayed, but you can view or hide each type.

*Generate a schema map*
How to generate a schema map.

1. Navigate to **System Definition > Tables & Columns**.
2. In the **Table Names** pane, select a table.
3. Click **Schema map**.

   The schema map for the selected table opens in a separate tab or window.

   You can print this map from the browser, if required.

*View the schema map*

The schema map shows the selected table in yellow, typically centered, and all tables related to that table, typically shown at the sides.
Figure 455: Schema Map v3 1

From this map:

- The check boxes at the top allow you to control which relationships to display. Select or clear a relationship type to display or hide tables with that relationship to the selected table.
- Each related table has a colored bar indicating the relationship to the selected table.
- You can point to the connector lines to display the details of a relationship between the two tables.

**Note:** Since relationships are shown as single lines for simplicity, the diagrams rendered are not entity relationship diagrams.

Using the Table Selector:

To view a schema map as a list, point to the table selector in the right corner:
You can:

- Click a table in the list to scroll the schema map to that table.
- Click the eye icon beside a listed table to hide or show that table in the schema map.
- Click the pin icon in the selector to keep the list open.

Using Related Tables:

Right-click a table node header to display a context menu with these functions:

- **Focus on this table**: make the selected table the new focus table and redraw the schema map based on the new selection.
  
  The new focus table is added as a breadcrumb at the top, allowing you to return to the previous table at any time.
- **Go to list**: display the list of records for the table.
- **Go to dictionary**: display the system dictionary, filtered for the selected table.

To hide a related table from view, click the eye icon in the node header (the node can be made visible again with the table selector).

For tables that are part of their own inheritance hierarchy, click the expand button (+) in the node header to add their inheritance hierarchy to the schema map.

**Viewing More Information**

Click the expand button (+) beside **Columns** to expand the table fields.
The reference fields show a red notation of the table they refer to.

If any tables extend from a table, their columns are displayed in reverse inheritance order. For example:

![Diagram showing extended table relationships]

**Figure 458: Schema Map Related Tables**

Here, the Server [cmdb_ci_server] table extends from Computer [cmdb_ci_computer], Hardware [cmdb_ci_hardware], and Configuration Item [cmdb_ci], and displays the columns from those tables.

Similarly, the Computer table displays the columns from the Hardware and Configuration Item tables.

**Tables module**

The Tables [sys_db_object] table contains a record for each table in the database.

To view the list of tables, navigate to **System Definition > Tables**. For example, you can filter the list to see extended table relationships.
Figure 459: Extended Table Relationships

To open the record for an existing table, click a table label. You can:

- View, add, or modify columns with a searchable and sortable embedded list, define the auto-number format, make the table extendable by other tables, and create modules for the table.
- Launch a schema map for a table by clicking the Show Schema Map related link.
- Open the dictionary entries for the table by right-clicking the form header and selecting Show Dictionary Record.
- Navigate directly to the default list or form view for the table by clicking the Show List or Show Form related link.
- Delete all records from a table by clicking the Delete All Records.
Unique record identifier

Each record in ServiceNow is identified by a unique 32-character GUID (Globally Unique ID) called a sys_id. The same sys_id value will never be generated twice, ensuring every record created in every table in every instance of ServiceNow in the world has a unique identifying value.

If two records have the same sys_id value, then one was copied to the other at the database level outside of the ServiceNow application. When created within the application, sys_id values are unique. The ServiceNow application and database should manage all operations on sys_id values. Typical end users do not see a record's sys_id and database administrators rarely use sys_id values.

**Note:** A sys_id of -1 is the sys_id of a new record. Once the record is inserted, it will be given a new sys_id.
**Globally unique record identifier**
Each record in ServiceNow is identified by a unique 32-character GUID (Globally Unique ID) called a sys_id.

The same sys_id value will never be generated twice, ensuring every record created in every table in every instance of ServiceNow in the world has a unique identifying value.

If two records have the same sys_id value, then one was copied to the other at the database level outside of the ServiceNow application. When created within the application, sys_id values are unique.

The ServiceNow application and database should manage all operations on sys_id values. Typical end users do not see a record's sys_id and database administrators rarely use sys_id values.

---

**Note:** A sys_id of -1 is the sys_id of a new record. Once the record is inserted, it will be given a new sys_id.

---

**Get the sys_id from the header bar**
Users can locate the sys_id of a record using the header bar.

1. Navigate to the record.
2. Right click the header bar and select **Copy URL**.

The sys_id is inside of the URL, after the parameter sys_id=. For example, the following is a URL for an Incident:

```
https://<instance name>.service-now.com/nav_to.do?uri=incident.do?sys_id=9d385017c611228701d22104cc95c371
```

Therefore the sys_id is 9d385017c611228701d22104cc95c371.

**Get the sys_id from a script**
Users can locate the sys_id of a record using a script.

- The sys_id value of a record can be found in a business rule (or any other server-side JavaScript) by dot-walking from the GlideRecord.

```
var id = current.sys_id;
```

- The sys_id of a record can be found in client-side JavaScript using `g_form.getUniqueValue()` as shown in the following example.

```
function onLoad() {
    var incSysid = g_form.getUniqueValue();
    alert(incSysid);
}
```

**Get the sys_id from the URL**
Users can locate the sys_id of a record by viewing the URL.

Since the sys_id of a record is always part of the URL for a link to that record, it is possible to retrieve the sys_id by viewing the URL.

- View the sys_id in the information bar of the browser by hovering over a link to the record.

For example, an Incident with the following URL: https://<instance name>.service-now.com/nav_to.do?uri=incident.do?sys_id=23dc968f0a0a3c1900534f399927740e, would have this sys_id: 23dc968f0a0a3c1900534f399927740e.
Unique index

A unique index on a table guarantees that the index key contains no duplicate values and therefore ensures that every row in the table is unique in some way.

Specifying a unique index makes sense only when uniqueness is essential for the data in a table.

You can view a table’s indexes as well as create your own index for a table using the table’s related list of indexes.

Create a table index

Create a custom index for a table.

Role required: admin

Constructing an effective index requires specialized knowledge in database architecture. We recommend that if you do not have this expertise, you consult someone who does.

Note: In this release, custom indexes are not tracked in update sets.

1. Navigate to System Definition > Tables.
2. In the list, find the table you want and click its label.
3. Navigate to the Database Indexes related list.
4. Click New.
5. In the Index Name field, type a name for your index.
6. Use the slush bucket to select the fields you want included in the index.
   The order in which you select the fields affects how the index works. If you do not have expertise in database design, we suggest you consult someone who does.
7. To create a unique index, check the Unique Index box.
8. Click Create Index.
   The Table Name field is there for your reference only. Overriding the default has no effect.

Tasks

A task is a specific type of record, controlled by the base class task. Within the platform, all ITIL processes are handled through tasks.

Tasks are created by users who are requesting the task to be performed, and are then updated as the task moves along the workflow. A task is created, work is performed upon it, and eventually it moves to a resolved state. Tasks allow users to request tasks, and track how they are being fulfilled by the appropriate parties. Tasks can be assigned to specific users or user groups. Many tasks are extended by child classes, such as Incident or Change.
Figure 461: A task in Record List view

Task table

Task [task] is one of the core tables provided with the base system.

It provides a series of standard fields used on each of the tables that extend it, such as the Incident [incident] and Problem [problem] tables. In addition, any table which extends task can take advantage of task-specific functionality for driving tasks.

The Planned Task plugin provides the Planned Task [planned_task] table, which extends the Task [task] Table to provide more fields for tasks to measure duration and effort.

Create a task

Tasks are not created directly on the task table. Instead, tasks are created on task child tables.

1. Click the New button on the task record list will launch the Task Interceptor.
2. Follow the on-screen instructions to locate the desired table that extends Task.
Define assignment rules

The instance can automatically assign a task to a user or group based on pre-defined conditions by using data lookup rules and assignment rules.

Assignment lookup rules example

In this example, the Data Lookup Plugin assignment lookup rule automatically assigns any incident with the Category of Request and Subcategory of Password Reset to Fred Luddy.

Figure 462: Assignment data lookup
Assignment rules module
The Assignment rules module allows you to automatically set a value in the assigned_to and assignment_group fields when a set of conditions occurs.

An assignment rule must also meet these additional criteria to run:

- The task record has been created or updated. Assignment rules do not apply to unsaved changes on a form.
- The task record must be unassigned. The record cannot have an existing value for either the assigned_to or assignment_group fields. Assignment rules cannot overwrite existing assignments (including assignments set by a default value or a previously run assignment rule).
- The assignment rule is the first rule that matches the table and conditions. If more than one assignment rule matches the conditions, only the rule with the lowest order value runs.

Condition editor example
In this example, the assignment rule uses a condition statement to automatically assign any incident opened in the Network category to the System Administrator in the Network assignment group.

![Assignment Rule](image)

Figure 463: Assignment

Data lookup rules
Data lookup rules offer a generic way to change any field value, not just assignment fields.

Data lookup rules offer the following improvements over the Assignment module:

- Ability to change any field value not just an assignment field
- More options to define when a rule runs:
  - On form change (Allows assignment rules to apply to unsaved changes on a form)
  - On record insert
  - On record update
- Option to replace existing values (including default values)

**Note:** You can define data lookup and Assignment rules at the same time. The system ignores any duplicate rules after an incident has been assigned unless you are using a data lookup definition option to replace existing values.

Precedence between assignment rules and business rules
When creating new assignment rules, keep in mind that business rules can take precedence over assignment rules in certain circumstances.

Assignment rules and business rules run in the following order:

1. All before business rules that run on a record insert with an order value less than 1000.
2. The first assignment rule with the lowest execution order and matching condition.
3. All before business rules that run on a record insert with an order value greater than or equal to 1000.
4. All after business rules that run on record insert.

**Workflow assignments**
An alternative to creating data lookup or assignment rules is to create one or more workflow tasks that assign a task record as part of a workflow.

Consider using a workflow for assignment if your process includes multiple steps or conditions such as requiring a particular group approve a request.

When using a workflow to manage task assignments, add a brief timer activity to the start of the workflow. Without this timer activity, the workflow runs before the parent record, the current record, is inserted into the database. After the timer activity completes, the workflow resumes using the parent record information from the database instead of the original current. Note that pausing a workflow in this way does not change a default workflow to a deferred workflow. For more information on how the workflow engine interacts with the database, see [workflow engine operation order](#).

**Baseline assignment rules example**
A baseline instance contains certain assignment rules.

### Table 382: Baseline instance rules

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Conditions</th>
<th>Assigned to user, Assigned to group, or Script</th>
</tr>
</thead>
</table>
| Networking | incident     | Category is Network                             | User: System Administrator  
|         |             |                                                 | Group: Network                             |
| Database or Software | incident     | Category is one of Request, Inquiry / Help, Software, Hardware | User: System Administrator  
|         |             |                                                 | Group: Software                             |
| SC Item fulfillment - Field Services | Ticket  | Parent.Task type is Request item | Group: Field Services |
| Release Planning | release_phase | Name is Plan                                   | Script: current.release.product.service.assigned_to; |
| IT Hardware | sc_req_item  | Approval is Approved and Item.Category is Request Computers and Hardware | User: System Administrator  
|         |             |                                                 | Group: Hardware                             |
| Service Desk | incident     | Active is true                                 | Group: Service Desk                         |

**Script example**
In this example, the assignment rules use JavaScript code to assign incidents to a particular assignment group based on the incident category.
Table 383: script incident category

<table>
<thead>
<tr>
<th>This incident category</th>
<th>Is assigned to this assignment group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>Hardware</td>
</tr>
<tr>
<td>Software</td>
<td>Software</td>
</tr>
<tr>
<td>Malware</td>
<td>Security</td>
</tr>
</tbody>
</table>

Figure 464: Assignment by script

The following script requires configuring the instance to add the Malware category and the Security assignment group.

```java
if(current.category=="Hardware"){
    current.assignment_group.setDisplayValue("Hardware");
} elseif(current.category=="Software"){
    current.assignment_group.setDisplayValue("Software");
} elseif(current.category=="Malware"){
    current.assignment_group.setDisplayValue("Security");
}
```

Assignment module rule

How to define an assignment rule.

1. Navigate to **System Policy > Rules > Assignment** and click **New**.
2. Complete the fields on the form (see table).
Figure 465: Assignment rule

Note: You might need to configure the form to see the fields.

Table 384: Assignment rule form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The descriptive name for the assignment rule.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active</td>
<td>An indicator of whether the assignment rule is active. Only active assignment rules take effect.</td>
</tr>
<tr>
<td>Applies to</td>
<td>The table with the records that the assignment rule applies to.</td>
</tr>
<tr>
<td>Table</td>
<td>Note: The list shows only tables and database views that are in the same scope as the assignment rule. If you select a custom table that extends the task table, you must clear the instance cache by navigating to <code>https://&lt;instance_name&gt;.servicenow.com/cache.do</code> in order for the assignment rule to work.</td>
</tr>
<tr>
<td>Conditions</td>
<td>The conditions in which the assignment rule will apply.</td>
</tr>
<tr>
<td>Assign to</td>
<td>The user which will be assigned the event.</td>
</tr>
<tr>
<td>User</td>
<td>The group which will be assigned the event.</td>
</tr>
<tr>
<td>Group</td>
<td>A script to determine advanced assignment rule functionality. The <code>current.variable_pool</code> set of variables is available.</td>
</tr>
<tr>
<td>Script</td>
<td>A script to determine advanced assignment rule functionality. The <code>current.variable_pool</code> set of variables is available.</td>
</tr>
<tr>
<td>Other fields</td>
<td>Choices are:</td>
</tr>
<tr>
<td>Match conditions</td>
<td>• Any - Assignment Rule will apply if any of the conditions are met.</td>
</tr>
<tr>
<td></td>
<td>• All - Assignment rule will apply if all of the conditions are met.</td>
</tr>
<tr>
<td>Order</td>
<td>A number to determine priority over conflicting assignment rules. If there are conflicting assignment rules, rules with lower Order values will take precedence over rules with higher Order values. If the Order values are all set to the same number the assignment rule with the first matching condition is run with precedence over the others. Only one assignment rule will run against a record, the first with a matching condition.</td>
</tr>
</tbody>
</table>
Create data lookup rules
How to define an assignment rule with Data Lookup and Record Matching Support.

1. Navigate to **System Policy > Rules > Assignment Lookup Rules**.
2. Click **New**.
3. Populate the assignment data lookup fields (see table).
4. Click **Submit**.

![Assignment Data Lookup](image)

**Table 385: Assignment data lookup fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Select the category the data lookup matches against.</td>
</tr>
<tr>
<td>Subcategory</td>
<td>Select the subcategory the data lookup matches against.</td>
</tr>
<tr>
<td>Configuration item</td>
<td>Select the configuration item the data lookup matches against.</td>
</tr>
<tr>
<td>Location</td>
<td>Select the location the data lookup matches against.</td>
</tr>
<tr>
<td>Assignment group</td>
<td>Select the assignment group to assign the incident to.</td>
</tr>
<tr>
<td>Assigned to</td>
<td>Select the user to assign the incident to.</td>
</tr>
<tr>
<td>Active</td>
<td>Set to Yes to run the rule or No to deactivate the rule.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter the order in which the rule runs compared to other rules on the same table. The Data Lookup Plugin runs the rule with the lowest order and matching values.</td>
</tr>
</tbody>
</table>

**Note:** The assignment lookup rule assigns incidents matching the values in the matcher fields (Category, Subcategory, Configuration Item, and Location) to the values in the setter fields.
(Assignment Group and Assigned To). A valid assignment lookup rule requires at least one matcher field and one setter field.

Important Task table fields

The Task table is a base class that provides fields for the core ITSM applications such as Incident, Problem, and Change Management. All applications that extend the Task table share these fields in common.

Table 386:

<table>
<thead>
<tr>
<th>Label</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>active</td>
<td>boolean</td>
<td>Specifies whether work is still being done on a task or whether the task's work is complete. By default, only application-specific business rules set the value of this field. For example, the incident autoclose business rule closes resolved incident that have not been updated for one day.</td>
</tr>
<tr>
<td>Additional comments</td>
<td>comments</td>
<td>journal_input</td>
<td>Displays and allows entry of comments about the task record. Each comment is inserted into the Activity field. For more information, see Journal Fields.</td>
</tr>
<tr>
<td>Approval History</td>
<td>approval_history</td>
<td>journal</td>
<td>Displays the history of approvals for the record. For more information, see Approvals.</td>
</tr>
<tr>
<td>Label</td>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Assigned To</td>
<td>assigned_to</td>
<td>reference</td>
<td>Specifies the user assigned to complete the task. By default, this field uses a <em>reference qualifier</em> to only display users with the itil role. Some applications override the parent reference qualifier to display relevant users. For example, the Service Order and Project Task tables filter users based on their skills.</td>
</tr>
<tr>
<td>Created</td>
<td>sys_created_on</td>
<td>glide_date_time</td>
<td>Displays the date and time when the task record was created.</td>
</tr>
<tr>
<td>Description</td>
<td>description</td>
<td>string</td>
<td>Displays and allows entry of a multi-line description of the work to be done.</td>
</tr>
<tr>
<td>Escalation</td>
<td>escalation</td>
<td>integer</td>
<td>Indicates how long the task has been open. Escalations are dynamically populated using service level agreements, which specify how long a task remains in each escalation state. Escalation states go from <em>Normal</em> to <em>Moderate</em> to <em>High</em>, and finally to <em>Overdue</em>. Record lists color code each task by escalation state.</td>
</tr>
<tr>
<td>Number</td>
<td>number</td>
<td>string</td>
<td>Displays an identifying number for each task record. This field is the display value for the Task table. The system generates this number when the task is created. To manage number generation, see <em>Number Maintenance</em>.</td>
</tr>
<tr>
<td>Label</td>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Opened</td>
<td>opened_at</td>
<td>glide_date_time</td>
<td>Displays the date and time when the task record was opened by a human for the first time.</td>
</tr>
<tr>
<td>Priority</td>
<td>priority</td>
<td>integer</td>
<td>Specifies how high a priority the task should be for the assignee. By default, this value is calculated by the calculatePriority business rule based on the <strong>Impact</strong> and <strong>Urgency</strong> values. Record lists color code each task by degree of priority.</td>
</tr>
<tr>
<td>Short Description</td>
<td>short_description</td>
<td>String</td>
<td>Displays and allows entry of a short description of the task, which is a human-readable title for the record.</td>
</tr>
<tr>
<td>State</td>
<td>state</td>
<td>Integer</td>
<td>Displays a choice list for status of the task:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Pending</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Open</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Work in Progress</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Closed Complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Closed Incomplete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Closed Skipped</td>
</tr>
<tr>
<td>Applications</td>
<td></td>
<td></td>
<td>Applications typically use Dictionary overrides on page 1444 to display application-specific states.</td>
</tr>
<tr>
<td>Sys ID</td>
<td>sys_id</td>
<td>GUID</td>
<td>Displays the <strong>unique record identifier.</strong></td>
</tr>
<tr>
<td>Label</td>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>--------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Task Type</td>
<td>sys_class_name</td>
<td>sys_class_name</td>
<td>Specifies the type of task, which corresponds to the child class. The system populates this value when a record is created on a child table. For more information on extending tables, see Tables and Classes.</td>
</tr>
<tr>
<td>Time Worked</td>
<td>time_worked</td>
<td>timer</td>
<td>Display a timer which measures how long a record is open in the form view.</td>
</tr>
<tr>
<td>Watch list</td>
<td>watch_list</td>
<td>glide_list</td>
<td>Specifies users who receive email notifications when the record is updated. By default, only the Incident, Change, and Service Catalog applications notify users listed in this field. For other Task-based applications, you must create custom email notifications sent to the users listed in this field.</td>
</tr>
<tr>
<td>Work notes</td>
<td>work_notes</td>
<td>journal_input</td>
<td>Displays and allows entry of comments viewable only by ITIL users. Each comment is inserted into the Activity field. For more information, see Journal Fields.</td>
</tr>
<tr>
<td>Work notes list</td>
<td>work_notes_list</td>
<td>glide_list</td>
<td>Specifies users who receive email notifications when work notes are added to the record. By default, only the Change, Problem, and Service Catalog applications notify users listed in this field. For other Task-based applications, you must create custom email notifications sent to the users listed in this field.</td>
</tr>
</tbody>
</table>
Journal fields

Journal fields work together to create a log of changes and comments as tasks are worked on.

Note: Journal fields work on audited tables only.

Fields of the journal_input type are multi-line text boxes which, upon save, add the comments into the Activity field with a notation.

Table 387: Fields that accept input into the journal

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional comments</td>
<td>Can be updated by any user.</td>
</tr>
<tr>
<td>Work notes</td>
<td>Can be updated by ITIL users.</td>
</tr>
</tbody>
</table>

These comments, as well as any changes to the record or email notifications sent out because of the record, are displayed in the activity formatter, which can be added to the form like a field.

Figure 466: Activity formatter for journal fields

Modify the task interceptor

You can modify the task interceptor.

1. Navigate to System Definition > Interceptors (this module may need to be activated).
2. Select the Task Interceptor.
   The Answers related list specifies what choices are presented and where the user is redirected after a choice is selected.
3. Modify the Answers related list as desired.
4. After making changes, test the interceptor by clicking **Try It**.

**Note:** To disable the task interceptor, rename it to something other than `task.do`. This disables it without deleting it.

---

### Out of box task fields

The table below lists out of box task fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>An identifying number for the task, which is used as the name of the task. Numbers are generated automatically when the task is created. Administrators can modify the number generation using number maintenance.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Priority</td>
<td>A signal of how high a priority the task should be for the assignee. When viewed on the Record List, the priority field will be color coded by degree of priority.</td>
</tr>
<tr>
<td>State</td>
<td>The status of the task, such as: Pending, Open, Work in Progress, Closed Complete, Closed Incomplete, and Closed Skipped.</td>
</tr>
<tr>
<td>Assigned To</td>
<td>The user or user group responsible with fulfilling the task.</td>
</tr>
<tr>
<td>Escalation</td>
<td>A signal of how long the task has been open. Usually, escalations are automatically configured using a Service Level Agreement. The longer the task is open, the more urgent the task is, moving from Normal to Moderate to High, and finally becoming Overdue. Like priority, this field is color coded in the list view.</td>
</tr>
<tr>
<td>Short Description</td>
<td>A short description of the task.</td>
</tr>
<tr>
<td>Task Type</td>
<td>A field which specifies the type of task, and adds a child class to extend the record.</td>
</tr>
</tbody>
</table>

**Planned task**

The Planned Task plugin provides a Planned Task [planned_task] table that extends the Task [task] table.

The Planned Task plugin cannot be activated independently. It gets activated when activating the Project Management plugin.

Planned tasks provide additional fields for tasks pertaining to time and effort as part of a planned, multi-stage process.

**Note:** If the Planned Task_v2 (com.snc.planned_taskv2) plugin is active when you upgrade to the Geneva or later releases, the system adds the Task column to the Planned Task [planned_task] table. The Task column is of type Composite Field, and stores the Short Description and Number of the task. During the upgrade, the system updates all records in the Planned Task table.

**Create a planned task**

Planned Tasks are not created directly on the Planned Task [planned_task] table.

Instead, planned tasks are created on planned task child tables. Clicking the New button on the Planned Task record list will launch the Planned Task Interceptor, which prompts the user to select a child table to create the planned task on:
To modify the planned task interceptor:

1. Navigate to System Definition > Interceptors (this module may need to be activated).
2. Select the Planned Task Interceptor.
3. The Related List Answers specifies what choices are presented, and where the user will be redirected to once they select the choice. Modify the list as desired.

Create a baseline

A Planned Task Baseline is a record of the planned task's start and end times at a particular moment in time.

To create a baseline, navigate to the top planned task's form and select the Create a Baseline related link:
To view the baseline, configure the related lists to add a related list of baselines:

![Baseline Table]

**Figure 468: Baselines tab**

The baseline can be viewed on a Gantt Chart using the related link.

*Measure time and effort*

The Planned Task `[planned_task]` table provides standard fields for tracking duration and effort.

Duration measures time from start to end date. Effort measures hours of work exerted on the project.

**Duration**

- Planned duration: the projected length of time for the planned task.
- Actual duration: the actual length of time so far for the planned task.
• Remaining duration: the Planned duration minus the Actual duration, which represents the projected length of time left.

Effort
• Planned effort: the projected amount of time that will be spent on the planned task.
• Actual effort: the actual amount of time that has already been spent on the planned task.
• Remaining effort: the Planned effort minus the Actual effort, which represents the project amount of work left.
• Percent complete: the Actual effort divided by the Planned effort, which estimates the percentage of planned work which has been completed.

Important planned task table fields
The following table contains a list of important Planned Task fields:

<table>
<thead>
<tr>
<th>Label</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual cost</td>
<td>work_cost</td>
<td>currency</td>
<td>The actual cost of the planned task, to be compared with the Estimated cost.</td>
</tr>
<tr>
<td>Actual duration</td>
<td>work_duration</td>
<td>glide_duration</td>
<td>The actual length of time (from start time to end time) of work on the planned task, to be compared with the Planned duration.</td>
</tr>
<tr>
<td>Actual effort</td>
<td>work_effort</td>
<td>glide_duration</td>
<td>The actual time spent working, to be compared to the Planned effort.</td>
</tr>
<tr>
<td>Critical Path</td>
<td>critical_path</td>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td>Estimated cost</td>
<td>cost</td>
<td>currency</td>
<td>An estimation of the cost of the planned task, to be compared with the actual cost.</td>
</tr>
<tr>
<td>HTML Description</td>
<td>html_description</td>
<td>html</td>
<td>A description field that accepts HTML markup.</td>
</tr>
<tr>
<td>Percent Complete</td>
<td>percent_complete</td>
<td>decimal</td>
<td>A percentage of the completed effort. Generated using the Planned effort and Actual effort fields.</td>
</tr>
<tr>
<td>Planned duration</td>
<td>duration</td>
<td>glide_duration</td>
<td>The estimated length of time (from start time to end time) of the planned task.</td>
</tr>
<tr>
<td>Label</td>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Planned effort</td>
<td>effort</td>
<td>glide_duration</td>
<td>The estimated amount of time spent working on the planned task.</td>
</tr>
<tr>
<td>Planned end date</td>
<td>end_date</td>
<td>glide_date_time</td>
<td>The estimated date and time for the planned task to end.</td>
</tr>
<tr>
<td>Planned start date</td>
<td>start_date</td>
<td>glide_date_time</td>
<td>The estimated date and time for the planned task to start.</td>
</tr>
<tr>
<td>Remaining duration</td>
<td>remaining_duration</td>
<td>glide_duration</td>
<td>The difference in planned and actual duration, representing the time left for the planned task.</td>
</tr>
<tr>
<td>Remaining effort</td>
<td>remaining_effort</td>
<td>glide_duration</td>
<td>The difference in planned and actual effort, representing the amount of work time left for the planned task.</td>
</tr>
<tr>
<td>Rollup</td>
<td>rollup</td>
<td>boolean</td>
<td>Read-only field managed by the system that identifies the task as having child tasks. A rollup task will have a number of its fields calculated from the children so those fields will be read-only.</td>
</tr>
<tr>
<td>Time constraint</td>
<td>time_constraint</td>
<td>string</td>
<td>A description of time constraints that apply to the planned task.</td>
</tr>
<tr>
<td>Top Task</td>
<td>top_task</td>
<td>reference (planned_task)</td>
<td>When different planned tasks are stacked in a hierarchy, this field populates with the highest-level parent task. For example, if Project A has a child Project B, and Project B has a child Project C, then Project C's Top Task is Project A. Project A's Top Task field will be blank.</td>
</tr>
</tbody>
</table>

*Planned task scripts*

A number of business rules and one script include power the dynamic calculation of crucial Planned Task fields.
Table 389: Planned Task Scripts

<table>
<thead>
<tr>
<th>Business Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Actual Work Start Value</td>
<td>Sets the planned task's Actual Start Date when State is set to the default work state.</td>
</tr>
<tr>
<td>Set Close Data on Inactive</td>
<td>Sets the planned task's close data when task becomes inactive.</td>
</tr>
<tr>
<td>Recalculate</td>
<td>Recalculates the planned task schedule fields when one of the schedule fields changes.</td>
</tr>
<tr>
<td>Auto close milestones</td>
<td>Automatically closes milestones when they are passed.</td>
</tr>
</tbody>
</table>

Planned task hierarchy

The Task Hierarchy tool available for Planned Task displays the relationship between parent and child planned tasks.

Out-of-box, the Task Hierarchy tool is available in both Project and Release Management.

Different Planned Task tables have different UI actions to launch the task hierarchy:

- To view a Project's hierarchy, navigate to the use the Task hierarchy context menu action.
- To view a Product's hierarchy in Release v2, navigate to the product and click the Product hierarchy related link.
- To view a Release's hierarchy in Release v2, navigate to the release and click the Release hierarchy related link.
The Task Hierarchy can be added to any planned task table by:

1. Navigating to **System UI > UI Actions**.
2. Selecting one of the existing Task Hierarchy UI Actions (e.g. Task hierarchy if Project Management is activated).
3. Change the table to the desired table and rename the UI Action if appropriate, and insert.

   The hierarchy should now be available as a UI Action on the new table’s form.

**Reminder table**

The Reminder [reminder] table provides a way to auto-generate reminders for a task.

The Reminder [reminder] table is one of the child tables of the Task [task] table. The Reminder [reminder] table can be used by any table that extends the Task table such as the Incident [incident] table.

Only the System Administrator can create or modify a Reminder table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Refers to the parent task record.</td>
</tr>
<tr>
<td>User</td>
<td>Specifies the user who has logged in to the system.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Remind me| Specifies the time before which the reminder must be sent. Available values are: **15 minutes, 30 minutes, 1 hour and 2 hours**. You can add values to this field as required.  
  
  **Note:** Internally in the system, the value is always converted to minutes and then stored in the Remind me field. |
| Before    | Refers to the date from the date field of the parent task record. This date and the time that you mention in the Remind me field are considered to send a reminder. Available values are: **Activity due, Due date, Follow up, and SLA due**.  
  
  **Note:** The Activity due field and the SLA due field are legacy fields with an unknown value that you cannot use. You can only use the Due date and Follow up fields. |
| Using     | Specifies the method by which the reminder must be sent. Available options are: **Send an Email** and **Outlook Calendar**.                          |
| Subject   | Specifies the subject or the reason of the reminder. Maximum number of characters allowed on this field is 100.                                  |
| Notes     | Provides a space where you can enter your comments regarding the reminder. Maximum number of characters allowed on this field is 8000.            |

**Task table modifications**

Modifications made to the Task table are applied to all child tables.

Be sure that the changes being made are appropriate for all of the child tables. Adding fields is a low-impact change, because the field can be hidden on tables that do not need it. However, deleting fields may cause unwanted data loss if the field is being used across tables.

**Note:** When adding choice list entries to a choice list on the Task table, make sure the entry value is unique.

You can use **dictionary overrides** to change some parts of a field definition in a way that does not to not apply to all child tables.

**Tasks workflow**

An administrator can specify a specific workflow process to apply to tasks that meet certain conditions.

Once a task is created that meets the conditions, the workflow will apply the process to the task, asking for approvals, notifying users, generating other tasks, running scripts, etc.

**Time cards**

The time card management feature works with the Task table to record time worked on Projects, Incidents, Problems, and Change Requests.

Task assignees can record time worked in the Time worked field on a task record or enter hours directly into their time card. Some tables support automatic time card creation based on start and end date fields.

Time cards also have an optional approval mechanism for project managers. Administrators and other roles that act as approvers can see all the time cards for the week. All users who are in a role that is responsible for working on tasks also can access their personal time cards. Time cards can have any of the following states.
Create a time card

Time cards can be created automatically or manually.

- Automatic: Configure time cards to be created when a user updates a task record. This behavior is controlled by a time card property that is set to false by default. See the table of properties in this page for details. In Incident, Problem, and Change records, the Time worked field must be added to the form.

  Note: The time card management plugin is required to use time cards. Some of the procedures on this page require the project management feature, which activates time cards automatically.

- Manual: Create a new time card for each task and enter the times manually.

  Users with the timecard_admin role can create a time card manually:

1. Navigate to Time Cards > All and click New.

   The Week starts on, State, and Category fields are completed automatically. The category defaults to Task work, but can be any of the following:
   - Task work
   - Admin
   - Meeting
   - KTLO (maintenance of existing system)
   - Out of office
   - Training

2. Select a Task from the pop-up list.

   This can be anything from the Task table.

3. Select your name from the list in the User field.

4. Click Submit.

   After the time card is created, the hours for that task can be incremented automatically from the Time worked field in the task record. This is controlled by a time card property, which is set to true by default. See the table of properties in this page for details. If automatic updates are not configured, the time card must be updated manually by the user or an administrator.
Managing time cards

The **My Time Cards > Current** module presents a page showing all of your time cards for the current week. There is also a control to Generate Task Cards. This button will search for all planned tasks that are scheduled for the current time card period, if you don't already have a time card for the task.

**Activate Time Card Management**

Administrators can activate the time card management plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

**Manage costs**

When the cost management feature is enabled, time cards can be used to manage the cost of labor in the Cost Management application.

Enable the cost management feature. When a time card is marked Approved, the user's rate (listed in the Labor Rate Card) is used to generate a one-time Expense Line for the time worked. If no Labor Rate Cards apply to the user, the property `com.snc.time_card.default_rate` defines a default rate.

**Roles**

The timecard_admin role enables users to approve, modify, and delete the time cards of other users.

**Record time worked**

Time accrued on a project or spent working on any record in the Task table is retrieved by the time card from the Time worked field.

This field is present on Project Task records by default, but does not appear on the Incident, Problem, and Change forms and must be added by personalizing the form. Time recorded in this field is used to populate an existing time card or to create a new time card if one does not already exist. This behavior is controlled by a time card property. The Time worked field has a counter that acts like a stopwatch for the duration of
the time spent in the record. The counter can be stopped and started by a button in the field. By default, the Time worked counter is enabled and begins recording the elapsed time when the record is opened. Stop the counter with the red stop sign button and restart it with the green play button.

Time counter started:

```
Time worked: 00:00:06
```

Figure 470: Time worked started

Time counter stopped:

```
Time worked: 00:00:26
```

Figure 471: Time worked stopped

If you are creating time cards from time worked entries, you can add the related list to display the time worked records on the time card form. You will also notice an informational message on the time card to let you know that changes to time worked records will override values in the time card. This is displayed using a formatter, which can be added or removed by configuring the form.

```
Time Card   ▼ = Required field

Time Worked property enabled, values may be overwritten by time worked records

Week starts on: 2011-01-02
State: Pending
Category: Task work
Task: INC0000003
User: John Roberts
```

Figure 472: Time worked notice

*Time card properties*
Users with the system administrator role can set time card properties by navigating to **Time Cards > Administration > Properties**.

Table 391: Time card properties table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.snc.time_card.autocreate</td>
<td>Auto-create a user’s time card when they update a task</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>com.snc.time_card.time_worked</td>
<td>Auto-fill a user's time card with time from their 'Time worked' entries</td>
<td>No</td>
</tr>
<tr>
<td>com.snc.time_card.update.effort</td>
<td>Update the task's 'Actual effort' based on the hours entered in the time card</td>
<td>No</td>
</tr>
<tr>
<td>com.snc.time_card.update.resource</td>
<td>Update the project/user's resource allocation record based on the hours entered in the time card</td>
<td>No</td>
</tr>
<tr>
<td>com.snc.time_card.start_day</td>
<td>What day should time cards start on, default is Sunday. Changing this value may create duplicate time cards for the week of the change, since time card queries are based on this value</td>
<td>Sunday</td>
</tr>
</tbody>
</table>

Tools for driving tasks

There are several tools available to drive tasks to completion.

These tools can be run on any table which extends Task.

- Approvals
  Approvals can be generated to a list of Approvers, either manually or automatically, according to Approval Rules. Approvals can be incorporated into workflows or can stand alone. For more information, see Process approvals on page 457.
  Approvals can be used on tables that do not extend Task.
- Assignments
  Assignment rules can automatically assign tasks to users or groups, ensuring that tasks are handled by the most appropriate team members. For more information, see Defining Assignment Rules.
- Service levels
  Service level agreements can track the amount of time that a task has been open, to ensure that tasks are completed within an allotted time.
- Inactivity monitors
  Inactivity monitors ensure that tasks do not fall by the wayside by notifying users when tasks have been untouched for a predefined period of time. For more information, see Setting Inactivity Monitors.
- Workflow
  An administrator can specify a specific workflow process to apply to tasks that meet certain conditions. After a task is created that meets the conditions, the workflow applies an automated process to the task. The process is defined in the graphical workflow editor.
Figure 473: Graphical workflow editor

As the process takes place, it updates any field designated as a workflow field.

Figure 474: Workflow field

Workflows are not specific to the tasks, but there are task-specific Workflow Activities (such as Task Activities and Approval Activities). For more information, see Workflow Overview.

Many to many task relations

By default, tasks can be related to each other using a parent/child relationship, such as a Problem with a group of child Incidents or a Catalog Request with a group of child Catalog Tasks.
However, it may prove useful to record exactly the nature of the relationship between the task records. When activated, the Many to Many Task Relations plugin allows administrators to define relationships between different tasks.

Request many to many task relations
How to request a plugin.

The Many to Many Task Relations plugin is included with the following plugins:

- Planned Task
- Field Service Management
- Project Management
- Governance, Risk, and Compliance

**Note:** Contact ServiceNow to activate the plugin by itself.

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
<tr>
<td>Specify the date and time you would like this plugin to be enabled</td>
<td>Date and time must be at least 2 business days from the current time.</td>
</tr>
</tbody>
</table>

**Note:** Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.

| Reason/Comments | Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows. |

3. Click **Submit**.

**Plugin manifest**
When the plugin is activated, the Task Relationships application is available with certain modules.

When the plugin is activated, the Task Relationships application is available with the following modules:
The following Relation Types are available by default:

- Knowledge Relation Types
  - Solution is documented in:
- Relationship Types
  - Caused by:
  - Contains:
  - Documenting Solution for:
  - Investigated by:
  - Permanent correction for:
  - Related to:
  - Requesting:
  - Solved by:

Define a relationship type

How to define a relationship type.

1. Navigate to Task Relationships > Relation Types and click New.
2. Populate the Parent Descriptor and Child Descriptor fields with a short description of the relationship between the two tasks, such as a parent descriptor of Caused By and a child descriptor of Causes.
3. Right-click the form header bar and select Save. The Name field automatically populates with the Parent and Child descriptors.
Define a task relationship allowed from the task relationship type record

It is possible to define Task Relationships Allowed from the Task Relationship Type record.

1. Scroll down to the related list and click **New**.
2. Populate the parent and child tables to define which tables are able to accept the relationship.
3. If desired, it is possible to define scripts to run in the Parent and Child script fields. These scripts are run when a parent or child record is run to automatically generate the other task (child or parent). These scripts use the current value of the new record, as opposed to the source record which triggered the script.
**View task relations**

How to view task relations.

The Task Relations field can now be added to the task form:

![Related Items](Image)

**Figure 476: Related items**

The plus icon can be used to add or remove tasks from the list:
Modify the displayed field
The fields displayed in the Task Relations field and editing interface are defined by the Reference Lookup’s list view.
To modify the displayed fields:

1. Navigate to a form that has a reference to the table whose display values you would like to modify within Task Relations field.
2. Click the magnifying glass to display the Reference Lookup list view.
3. Right-click the list header and select Configure > List Layout.

The selected List layout will be used as the display value for records within the Task Relations field.

**Mark as Solution button**

This button gets added to the KB popup view and is displayed when you search the knowledge base from a task record.

Clicking the button creates a record in the Task / KB Relationships [task_rel_kb] table to associate the KB article with a task. Its functionality has been replaced in the KCS plugin and is no longer necessary. You can disable this button by marking the active field false on the 'solution_button' UI macro.

**Task table UI actions**

Once the task relationships are defined, it is possible to use UI Actions to define the task relationship as a new task is being created from an old task.

Below are a few examples.

---

**Warning:** These examples may not work on all instances. They are provided as illustrative examples.

---

**UI Actions examples**

**Cause an incident**

This UI Action allows the change management team to log an incident directly from the change request and records that the incident was caused by the change.

Create a new UI Action on the Change Request [change_request] table and place the following into the script:

```javascript
var inccaus = new GlideRecord("incident");
inccaus.short_description= current.short_description;
inccaus.comments= current.comments.getHTMLValue();
// inccaus.parent = current.sys_id;
inccaus.insert();
CauIncident();
gs.addInfoMessage("Incident " + inccaus.number+" created");
action.setRedirectURL(current);
action.setReturnURL(inccaus);

function CauIncident(){
    var m2m =new GlideRecord('task_rel_task');
m2m.initialize();
m2m.child= current.sys_id;
m2m.parent= inccaus.sys_id;
m2m.type.setDisplayValue("Caused by::Causes");
m2m.insert();}
```

**Cause a problem**

This is a UI Action to allow the change management team to record a problem from a change request and record that the problem was caused by the change.
Create a new UI Action on the Change Request [change_request] table and paste the following script:

```javascript
var probcaus = new GlideRecord("problem");
probcaus.short_description= current.short_description;
probcaus.comments= current.comments.getHTMLValue();
// probcaus.parent = current.sys_id;
probcaus.insert();
CauProblem();

gs.addInfoMessage("Problem " + probcaus.number+" created");
action.setRedirectURL(current);
action.setReturnURL(probcaus);

function CauProblem(){
var m2m = new GlideRecord('task_rel_task');
m2m.initialize();
m2m.child= current.sys_id;
m2m.parent= probcaus.sys_id;
m2m.type.setDisplayValue("Caused by::Causes");
m2m.insert();}
```

**Fix a problem**

This UI Action allows a change request to be generated from a problem, recording that the change will fix the problem.

Create a new UI Action on the Problem [problem] table, and paste the following code:

```javascript
var fixchg = new GlideRecord("change_request");
fixchg.short_description= current.short_description;
fixchg.comments= current.comments.getHTMLValue();
// fixchg.parent = current.sys_id;
fixchg.insert();
FixChange();

gs.addInfoMessage("Change " + fixchg.number+" created");
action.setRedirectURL(current);
action.setReturnURL(fixchg);

function FixChange(){
var m2m = new GlideRecord('task_rel_task');
m2m.initialize();
m2m.child= current.sys_id;
m2m.parent= fixchg.sys_id;
m2m.type.setDisplayValue("Fixes::Fixed by");
m2m.insert();}
```

**Using special characters in an XML file**

The XML specification defines five predefined entities that represent special characters, and requires that all XML processors honor them. If these characters are used in a password, you will experience unexpected results.

The most likely scenario to experience this conflict is when you set the password for a MID Server.

The following characters represent the five pre-defined entities:

- "
- &
- '
XML and Special Characters

If you use the pre-defined entity characters in an XML file, such as the MID Server configuration file, you need to encode them.

To encode pre-defined entities into an XML document:

- replace " with &quot;
- replace & with &amp;
- replace ' with &apos;
- replace < with &lt;
- replace > with &gt;

For example, to specify the password as test& in the MID Server config.xml file:

```xml
<parameter encrypt="true" name="mid.instance.password" value="test&amp;"/>
```

System clone

The System clone application allows users with the clone_admin or admin role to clone data from one instance to another.

This functionality is primarily used to clone a production instance over an existing sub-production instance before developing or testing changes. All clones are performed using the most recent nightly backup. This application applies to instances in Gen2 and later datacenters.

Figure 478: Clone process
Clone from a backup

The platform uses data from the most recent nightly backup of the source instance when cloning. Backups used for cloning are at most 36 hours old.

After cloning from a backup, the target instance is unavailable for several minutes before the clone is marked as complete in the source instance. If the source and target instances are on different versions of the ServiceNow platform, the target instance is modified to match the source instance version during this time.

When starting a clone from a backup, the date and time the backup was taken, as well as periodic progress messages, appear in the Clone Log related list.

Figure 479: System clone backup log

Clone over production instances

Production instances cannot be used as the target instance for a clone after the instance is live. Production clones are created during non-core business hours. Tables are backed up sequentially rather than simultaneously. Modifying data on the source instance during a clone can cause a data mismatch between records or duplicate record entries. This issue is minimized by running a clone after normal business hours.

When scheduling a clone for a production instance, the system automatically follows this process:

- Determines the instance region.
- Determines the non-core business hours for the region.
- Restricts the possible cloning time that can be specified on the Clone Request form to the non-core hours.

<table>
<thead>
<tr>
<th>Region</th>
<th>Non-core business hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Americas East Coast</td>
<td>23:00 to 04:00 Eastern time (UTC –5) and all day Saturdays and Sundays</td>
</tr>
<tr>
<td>Americas West Coast</td>
<td>20:00 to 01:00 Pacific time (UTC –8) and all day Saturdays and Sundays</td>
</tr>
<tr>
<td>Region</td>
<td>Non-core business hours</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>Australia</td>
<td>23:00 to 04:00 Australian Eastern time (UTC +10) and all day Saturdays and Sundays</td>
</tr>
<tr>
<td>Europe Amsterdam</td>
<td>00:00 to 05:00 Central European time (UTC +1) and all day Saturdays and Sundays</td>
</tr>
<tr>
<td>Europe London</td>
<td>23:00 to 04:00 British time (UTC) and all day Saturdays and Sundays</td>
</tr>
<tr>
<td>Asia Hong Kong</td>
<td>23:00 to 04:00 Hong Kong time (UTC +8) and all day Saturdays and Sundays</td>
</tr>
<tr>
<td>Asia Singapore</td>
<td>23:00 to 04:00 Singapore standard time (UTC +8) and all day Saturdays and Sundays</td>
</tr>
</tbody>
</table>

Use System clone

The System clone application automates much of the cloning process. Specific tasks are performed during the clone request process.

An administrator defines what data is preserved on the target instance and what data from the source is not cloned, then initiates the clone process. A background job ensures that all cloning requirements are met and carries out the clone process. These processes and procedures are performed on the source instance.

During a clone, the target instance may be intermittently unavailable. After clone completion, you have up to 24 hours to contact ServiceNow Customer Support and request a rollback of the target instance to its pre-clone state. You are notified when the rollback is complete.

Clone request process

In response to a clone request, the ServiceNow platform performs the following tasks.

1. Generates a file to preserve operational data on the target server.
   This file contains the data preserved by data preservers.
2. Copies the database schema from the source instance to the target instance.
3. Creates tables in the target instance database using the source instance table definitions.
4. Copies data from the most recent nightly backup of the source instance to the target instance database.
   Certain large tables are normally excluded. These include audit, log, and email tables.
5. Briefly disables UI traffic and requests to the target instance server.
6. Displays the message **Clone in progress...** to any user accessing the target instance.
7. Restores operational data preserved from the target instance.
8. Runs any post-clone cleanup scripts on the target instance.
9. Briefly suspends all email functions on the target instance.
10. Queues an event to regenerate text indexes.
11. Enables UI traffic and requests to the target instance server.

Create a clone target
You create a target instance record before the instance can be selected as clone target.
Role required: clone_admin or admin

Note: If you are using IP Access Controls on your instance, the target instance must allow the IP range 10.0.0.0/10.255.255.255 to communicate on a local network to allow the clone. For more information, see IP range based authentication on page 2479.

1. Navigate to System Clone > Clone Targets.
2. Click New.
3. Enter the URL for the receiving instance (target).
4. Enter the basic authentication credentials for a user account with the admin role on the target instance.

Note: These credentials must exist in the User [sys_user] table as a user record or as part of an LDAP integration. Clone requests cannot use single sign-on to authenticate users from an identity provider.

5. Click Submit.
   The system checks connectivity and validates the user credentials against the target instance.

Start a clone
For most instances, an administrator can start a clone.

Role required: clone_admin or admin

For instances that use an Oracle database, see KB0538884 - System Clone Support for Oracle Customers.

1. Navigate to System Clone > Request Clone.
2. Select a Target instance to receive the cloned data.
3. Complete the Options form section.
Table 393: Clone options

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone from</td>
<td>Select the source database, if more than one database is available. If only one source is available, the <strong>Clone from</strong> menu does not appear. By default, the clone process ignores this field and uses data from the most recent nightly backup of the source instance. The clone process only uses this field value if the clone from backup fails for some reason.</td>
</tr>
<tr>
<td>Exclude audit and log data</td>
<td>Select this option to prevent the cloning of tables specified in <strong>System Clone &gt; Exclude Tables</strong>. This option is selected by default. Clearing the check box causes the platform to ignore the <strong>Exclude Tables</strong> module.</td>
</tr>
</tbody>
</table>
| Exclude large attachment data | When selected, the clone excludes any record in the Attachment [sys_attachment] and Attachment Document [sys_attachment_doc] tables that meet these criteria.  
  • The table_name value does not start with ZZ_.  
  • The table_name value is not sys_certificate, ecc_agent_jar, ecc_agent_mib, or sys_store_app. |
| Preserve theme             | Select this option to prevent the cloning of CSS elements, colors, and banner displays. This option enables or disables any data preservers where **Preserve theme** is **True**.                                                             |

4. Schedule a **Date and time** to perform the clone.  
   Scheduling prevents multiple clones from occurring on the same target instance concurrently. A clone must be scheduled at minimum four hours in advance.

5. Enter an **Email** address to receive alerts after the clone finishes, is canceled, or has an error.
6. Click **Submit**.

7. In the authentication window that appears, enter the **Username** and **Password** for an administrator account on the target instance.

8. Click **Authenticate**.

9. Review the clone settings and click **OK**.
   An email is sent to the supplied address after the clone finishes, is canceled, or has an error.

**Warning:** If the clone from backup fails for some reason, the clone process fails over to the legacy clone engine. The legacy clone engine cannot preserve data from extended tables, relationships, hierarchies between tables, and dot-walked queries. You may want to reschedule a system clone or manually transfer data in such cases.

---

**Post-clone cleanup scripts**

The Cleanup Scripts module allows you to define scripts to automatically run on the target instance after a clone is finished.
Cleanup scripts run after data preservers and the clone is complete. You can add new post-cloning scripts on the source instance to perform any action that can normally be accomplished through script includes or business rules. The following post-clone cleanup scripts perform various actions on the target instance.

**Table 394: Post-clone cleanup scripts**

<table>
<thead>
<tr>
<th>Script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable emails</td>
<td>Disables email on the target instance. A default data preserver maintains other email settings from the target instance.</td>
</tr>
<tr>
<td>Regenerate text indexes</td>
<td>Rebuilds text indexes on the target instance after a clone. Text indexes are not cloned from the source to the target instance.</td>
</tr>
<tr>
<td>Clear scheduled job node association</td>
<td>Resets any scheduled jobs that were active on the source instance to the Ready state. This script also clears the value of the System ID and Claimed by fields on all scheduled jobs.</td>
</tr>
<tr>
<td>Install deactivated plugin</td>
<td>Enables the Domain Separation plugin for instances that use this feature.</td>
</tr>
<tr>
<td>Drop backup tables</td>
<td>Schedules the deletion of the data contained in the target instance database prior to the clone. This original data is preserved for 24 hours following a clone to allow you to roll back an instance to the pre-clone state. If the target instance is downgraded as part of the clone, backup data is not available.</td>
</tr>
</tbody>
</table>

**Work with clones**

You can add a system property to allow an instance to be cloned, cancel a clone, and send clone status notifications.

**Allow a clone**

Production instances cannot be used as the target instance for a clone after the instance is live. Also, the glide property glide.db.clone.allow_clone_target must be set to **True** to allow an instance to be a clone target. Attempting to clone over an instance where this property is **False** displays the following error:

*Figure 480: Clone target invalid*

This property is **False** by default, and to **True** on instances where the name ends in Dev, Test, Stage, UAT, or QA.

To modify the value of this property, *add the property* to the [sys_properties] table.
Cancel a clone

You can cancel a clone without negatively impacting system stability or usability. Canceling a clone restores the target instance to the pre-clone state, retaining all original data.

To cancel a scheduled or in-progress clone, navigate to System Clone > Clone History. Select the clone and click Cancel Clone.

Send clone status notifications

You can configure the instance to send notifications regarding an active clone. These notifications indicate when the clone finishes, is canceled, or has an error. Use the Email upon completion field to specify who to send status notifications to. Users specified as the primary or secondary support contacts for your company also receive these notifications.

Clone to an instance on a different version

The System Clone application can target an instance running a different ServiceNow version from the source.

A central web service controls clone processing and automatically modifies the target instance version to match the source instance version. This matching process starts up to eight hours before the time specified in the Date and time field on the System Clone form. This web service also ensures that there is enough disk space on the target instance for the clone to proceed.

When cloning from a backup, the target instance does not need additional time to upgrade or downgrade. The ServiceNow platform performs any version changes during a brief window where the target instance is unavailable, after it copies data from the source instance backup.

View clone history and active clones

You can view clone history information. All previously and currently scheduled clones from a particular instance display on this table. The clone history module additionally lists clones that have begun but have not yet finished.

Role required: clone_admin or admin

Navigate to System Clone > Clone History.

Clone history also displays the State for current and past clones. Clones in the draft state do not appear on the clone history table.
Table 395: Clone state

<table>
<thead>
<tr>
<th>Clone state</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requested</td>
<td>The clone was requested and is awaiting approval.</td>
</tr>
<tr>
<td>Scheduled</td>
<td>The clone is ready to begin at the scheduled time and date.</td>
</tr>
<tr>
<td>Active</td>
<td>The clone is currently running.</td>
</tr>
<tr>
<td>Completed</td>
<td>The clone completed successfully.</td>
</tr>
<tr>
<td>Canceled</td>
<td>A user canceled the request.</td>
</tr>
<tr>
<td>Hold</td>
<td>The server rejected the clone request. This can happen either because the clone was not ready to proceed by the scheduled time or because additional clone requests were submitted before the first one completed.</td>
</tr>
<tr>
<td>Error</td>
<td>The clone encountered an error while running. Contact technical support for help resolving this issue.</td>
</tr>
</tbody>
</table>

Figure 481: Active system clone

After starting a clone, the Clone Log and Database Table Clones related lists appear on the form. These related lists show general log messages, and the details of individual tables respectively.

The duration of time a clone remains active varies depending on the amount of data being cloned, and whether the source and target instance are in the same physical location. If a clone takes longer than anticipated, ServiceNow Technical Support can identify additional details about the clone progress.
Create a data preserver

Data preservers maintain certain data on the target instance. Sometimes, preserving certain data on a target instance is desirable. For example, when using a MID Server, you can avoid overwriting the MID Server [ecc_agent] table.

Role required: clone_admin or admin

Preserved data is stored in a dynamically generated list on the target instance before the clone and restored on the target instance after the clone is complete. You define data preservers on the source instance.

Data preservers are primarily intended to preserve system settings and themes, such as instance-specific authentication settings. Do not use data preservers to transfer large sets of data, such as user groups. If you must preserve table data such as users, groups, and roles, consider exporting the records to a file and importing it after the clone is complete.

Consider whether to preserve the data in the following tables.

- Bookmark [sys_ui_bookmark]
- Recent Selection [sys_ui_recent_selection]
- User Preference [sys_user_preference]

1. Navigate to System Clone > Preserve Data.
2. Click New.
3. Enter the table label as the Name, for example, User Preference for the [sys_user_preference] table.
4. Select the Table to be preserved.
5. Select the Theme check box if the data being preserved is a UI property.
6. Define the data to be preserved using the Condition builder on page 157.

You can use conditions to define particular records you want to preserve during a clone. For example, to only preserve particular system properties, you can add conditions for each property name you want to preserve.
Warning: If the clone from backup fails for some reason, the clone process fails over to the legacy clone engine. The legacy clone engine cannot preserve data from extended tables, relationships, hierarchies between tables, and dot-walked queries. You may want to reschedule a system clone or manually transfer data in such cases.

7. Click Submit.

Delete a data preserver
You can delete data preservers that you no longer need. The platform protects certain essential data by default.

Role required: clone_admin or admin

Do not modify or delete the following data preserver records:

- Core Instance Properties
- Semaphores
- Email Accounts

1. Navigate to **System Clone > Preserve Data**.
2. Select the data preserver to delete.
3. Perform either of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To delete a condition</td>
<td>Click the X icon beside the condition, and then click <strong>Update</strong>.</td>
</tr>
<tr>
<td>To delete the entire data preserver record</td>
<td>Click <strong>Delete</strong>, and then click <strong>Delete</strong> in the confirmation prompt.</td>
</tr>
</tbody>
</table>

**Exclude a table from cloning**

You can exclude a table to create an empty but usable table on the target instance.

Role required: clone_admin or admin

The **System Clone > Exclude Tables** module lists the tables that are not copied during a system clone. By default, the system excludes tables for logging, auditing, notifications, and license usage.

The system cannot exclude tables that extend the Task table and that are also flattened into it as part of the **table per hierarchy extension model**. Since these extended tables are part of the same physical database table, the system clones the data when it clones the Task table.

You can exclude tables that extend the Task table under two conditions. Either the system stores the tables in their own physical tables as part of the table per class extension model, or you exclude the Task table itself.

To preserve the existing data on the target instance, see preserving data. Data on tables that reference the table, such as business rules, is not excluded.

1. Navigate to **System Clone > Exclude Tables**.
2. Click **New**.
3. Enter the table **Name**.
4. Click **Submit**.

**Data preservation and multiple provider single sign-on**

The system automatically creates the necessary data preservers for cloning when you activate Multiple Provider Single Sign-On integration.

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate</td>
<td>X.509 Certificates [sys_certificate]</td>
<td>None</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Conditions</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Core Instance Properties | System Property [sys_properties] | • [OR] [Name] [is one of] glide.authenticate.external, glide.authenticate.external.logout_redirect  
• [OR] [Name] [starts with] com.snc.integration.saml_esig  
• [OR] [Name] [is one of] glide.smtp.port, glide.smtp.auth, glide.smtp.encryption  
• [OR] [Name] [starts with] glide.authenticate.multisso  
• [OR] [Name] [is] glide.authenticate.sso.redirect.idp |
| Digest Properties        | Digest Properties [digest_properties] | None                                                                                                                                 |
| Identity Providers       | Identity Providers [sso_properties] | None                                                                                                                                 |
| SAML2 Update1 Properties | SAML2 Update1 Properties [saml2_update1_properties] | None                                                                                                                                 |

**Note:** Although you can modify these data preservers, a good practice is to avoid changing them. The Digest Properties [digest_properties], Identity Providers [sso_properties], and SAML2 Update1 Properties [saml2_update1_properties] tables are required for multiple source single sign-on to function properly. If multiple source single sign-on is disabled on the target instance, you can safely remove all three data preservers. Remove them at the same time, as the system terminates the clone with an error message when you attempt to clone with one or two of these tables being preserved.

**Clone an instance with a SAML integration**

Preserving SAML SSO-related settings can prevent the target instance from redirecting all authentication requests to the original IdP with the wrong issuer and audience parameters.

To preserve SAML settings, create data preservers for the following tables:

- System Property [sys_properties]
- X.509 Certificates [sys_certificate]
- User [sys_user]

**Preserve SAML properties**

For a clone target instance to keep its existing SAML integration, you must edit the **Core Instance Properties** data preserver to include the SAML properties.

Role required: admin

1. Navigate to **System Clone > Preserve Data**.
2. Select **Core Instance Properties**.
3. Add the following **Conditions**.
   - [OR] [Name] [is one of] glide.authenticate.external, glide.authenticate.external.logout_redirect, glide.authenticate.failed_requirement_redirect
• [OR] [Name] [starts with] [glide.authenticate.sso.saml2]
• [OR] [Name] [starts with] [com.snc.integration.saml_esig]
Clone Data Preserver - Core Instance Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Core Instance Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theme</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>System Property [sys_properties]</td>
</tr>
</tbody>
</table>

**Conditions**

- **Name** is `.glide.sys.schedulers`
- **Name** starts with `glide.db`
- **Name** is `glide.ui.max.transaction`
- **Name** starts with `glide.email`
- **Name** starts with `glide.op3`
- **Name** is `instance_id`
- **Name** is `instance_name`
- **Name** starts with `glide.installation`
- **Name** is one of `glide.authenticate.external`, `glide.authenticate.external.logout_redirect`, `glide.authenticate.failure_requirement_redirect`
- **Name** starts with `glide.authenticate.ssos`
- **Name** starts with `com.snc.integration.san`
Note: Ensure the Theme check box is cleared so these properties are preserved regardless of whether you preserve the instance theme.

4. Click Update.

Preserve SAML certificates
Preserve the SAML 2.0 certificate in the target instance so the SAML integration to continue working after a clone.

If you activated Multiple Provider Single Sign-On, the system has already created a data preserver for this table. Verify that it is correctly configured.

1. Navigate to System Clone > Preserve Data.
2. Click New.
3. Enter the following field values.

Table 397: Data preserver field values

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>X.509 Certificate [sys_certificate]</td>
</tr>
<tr>
<td>Conditions</td>
<td>[Name] [starts with] [SAML 2.0]</td>
</tr>
</tbody>
</table>

4. Click Submit.

The system clone now preserves the SAML certificates.

Preserve SAML users
If the source instance does not include the SAML users that you log in with, create a data preserver for those users.

1. Navigate to System Clone > Preserve Data.
2. Click New.
3. Enter the following field values.

Table 398: Data preserver field values

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>User [sys_user]</td>
</tr>
<tr>
<td>Conditions</td>
<td>Enter the users you want to continue to log in from SAML.</td>
</tr>
</tbody>
</table>

4. Click Submit.

Integration with third-party applications and data sources
Integration is the exchange of information between company applications and a ServiceNow instance. ServiceNow integrates with many third party applications and data sources.

The most common integrations are with CMDB, Incident Management, Problem Management, Change Management, User Administration, and Single Sign-on.
A variety of techniques can be used, most notably Web Services, JDBC, LDAP, Excel, CSV, and Email, as well as any industry standard technologies that use SOAP, REST, or WSDL. Additionally, API and command-line integrations can be done using a MID Server.

**Note:** You can establish integrations before or after the ServiceNow implementation goes live, however the best practice is to enable integrations before your implementation goes live. Enabling integrations prior to going live allows you to test the integration. This is especially important for integrations to user lists or single sign-on applications since a problem in one of these integrations can prevent users from logging on.

### Technologies

The ServiceNow platform is based on service-oriented architecture (SOA), in which all data objects can use web services to access bi-directional data-level integration. The interface is also direct and dynamic because all modifications to existing objects and all new objects are automatically published as a Direct Web Service. A more indirect web service creation and usage can be achieved through Mapped Web Service where a transform map is used to gather incoming web service data into the final targeted tables. Finally, an advanced Scripted Web Service technique is available for defining process-based web services, where data is irrelevant, but serves more as a trigger for a process or a composite of actions that execute at the server.

Additionally the platform offers a rich interface for loading external data using import sets. Using this feature, you can load from various data sources such as HTTPS, FTPS, and SCP using file formats such as XML, CSV, and Microsoft Excel XLS files. Information can also be pulled from a data source using a direct JDBC connection, provided the network connectivity allows.

Information can be pulled from the platform to an external platform using an **ODBC Driver**.

Forms, lists, and reports on the platform can be accessed directly using a URL, which facilitates integration on the UI level between two or more web applications.

A handful of single sign-on technologies is identified and implemented out of the box to allow fast integration with your portal, however, the technique is customizable in a script to allow for flexibility in the different SSO environments our customers have.

### Integration between ServiceNow Instances

There are times when you find you need to perform a specific integration between your instance and another ServiceNow instance. Instance-to-Instance integrations are a snap because all of the integration points exist between the two instances.

Update Sets also provide exchanges between instances. For more information, see *Getting started with Update Sets*.

### User interface integration

Integration at the user interface level allows standard, seamless embedding of ServiceNow charts and reports into portals, exposure of knowledge base articles into web sites, and the embedding of in-context URLs into other web enabled applications.

- Integrating ServiceNow with your Intranet (Web IFrame)
- Launch in-context
Integration options

Nearly all ServiceNow customers obtain additional value by integrating with third-party applications. ServiceNow customers have the following options for integrating with third-party applications.

- Activate a ServiceNow provided integration.
- Install a certified integration from the ServiceNow Store.
- Install an integration from Share.
- Contact your sales representative for available custom-built integrations.
- Build your own custom integration using the platform's integration interfaces.
- Use the Technology Partner program, which provides ServiceNow customers with certified integrations with third-party applications. Certification indicates that the integrations have passed a set of interoperability, security, and performance test criteria as defined and witnessed by ServiceNow.

See Technology Partner program for more information.

Note: Certified integrations have passed a set of interoperability, security, and performance test criteria defined by ServiceNow.

Login integration

Integrations that support a user logging into ServiceNow include techniques used for single sign-on, dynamic user record creation, as well as loading users, groups and location data.

Single sign-on

Bypass the ServiceNow login authentication by allowing a user of your corporate portal to automatically login to the system with centrally managed roles and authorization.

- External Authentication
- Intranet and Login Integration

User data

Using our integration with LDAP or secure LDAP, dynamically create user accounts by importing from your Active Directory or OpenLDAP implementation. You can do this securely via a point-to-point VPN. Additionally, you can use the dynamic mapping of data sources such as files or databases via JDBC, to import user, groups, and location data into ServiceNow using import sets.

- LDAP Integration
- LDAP Import Map
- Import Sets

Data and process integration

Data and process integration can be categorized as either event based, or batch load.

An event-based integration, more on-demand and usually with a smaller payload, creates records for the purpose of triggering additional workflow in the system. Event based integration records usually go through an external queueing table called the ECC Queue.
Batch or data load based integration is typically done with a scheduled import set, where mappings and script can be defined to gather data from various data sources into the system and transform them into records in tables.

**Event-based options**

- Web Services
- External Integration Using JMS
- MID Server

**ServiceNow provided integrations**

ServiceNow provides many integrations as part of the platform. These integrations are considered part of the platform and are provided at no additional charge.

**Table 399: Provided Integrations**

<table>
<thead>
<tr>
<th>Integration</th>
<th>Type</th>
<th>Integration Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altiris (version 2.0)</td>
<td>MID Server</td>
<td>CMDB</td>
</tr>
<tr>
<td>Google Maps</td>
<td>Web services</td>
<td>Varies</td>
</tr>
<tr>
<td>Google Custom Search</td>
<td>Web services</td>
<td>Knowledge</td>
</tr>
<tr>
<td>Microsoft SMS / SCCM</td>
<td>MID Server</td>
<td>CMDB</td>
</tr>
<tr>
<td>Verizon eBonding</td>
<td>Web services</td>
<td>Incident</td>
</tr>
</tbody>
</table>

**Altiris integration 2.0**

The Altiris integration is a one direction import of the Altiris data into ServiceNow. CMDB (Configuration management database). The integration keeps the ServiceNow CMDB up to date with Altiris SQLServer database.

**Google Maps API support**

Map pages enable you to graphically display data on a Google map page based on location data.

**Microsoft SMS/SCCM integration**

The Microsoft SMS/SCCM integration is a one direction import of SMS/SCCM data into ServiceNow’s Configuration Management Database (CMDB).

**Migrate the Verizon eBonding Integration to a Production System**

This topic outlines the tasks required for moving the Verizon eBonding integration from a Dev/Test environment to a Production environment.

**Altiris integration 2.0**

The Altiris integration is a one direction import of the Altiris data into ServiceNow CMDB (Configuration management database). The integration keeps the ServiceNow CMDB up to date with Altiris SQLServer database.

**Note:** Functionality described here requires the Integration - Altiris 2.0 plugin.
Data Import

Relevant data is imported from the Altiris database to the CMDB. The Altiris database is not written to, it is considered an authoritative source. The import is achieved using a JDBC connection via the MID Server.

Figure 482: Altiris integration application

Information pulled from Altiris includes:

- PC Hardware (Model, Manufacturer, memory, clock speed, number of CPUs, etc.)
- Operating System Information (Name, Service Pack)
- Printers
- Disk information (physical, network, and logical)
• Network (IP Address, Netmask)

**Configuration and Operational Modules**

Enabling this integration will create the **Integration - Altiris** application.

The following are the configuration and operational modules for this integration.

**Setup**

- Configure the data sources from one form
  - Provide the Database Server IP Address
  - Provide the Database Name
  - Provide the Database User ID and Password (this will need to be created on SQL DB side, see this article for help with this step: [http://technet.microsoft.com/en-us/library/aa337562.aspx](http://technet.microsoft.com/en-us/library/aa337562.aspx)
  - Find and select the MID Server
- Specify Database server settings and MID server
- Test configuration

**Scheduled Import**

- Schedule the execution of the import or import immediately

**Data Sources**

- A list of the pre-configured data sources defining the external CMDB database

**Progress**

- A historical list of progress on scheduled imports

**Transform History**

- A historical list of transformations performed during scheduled imports

---

**Warning:** If you have activated an existing integration of the previous version:

1. Activating the 2.0 plugin does not "add to" or "remove" anything from the existing integration.
2. If transitioning from the old integration to this new one, considerations need to be given to customizations already done eg. mapping enhancements or using different coalesce values, these will have to be re-implemented.
3. Both plugins could run at the same time, provided data is coalescing the same way - until there is no need for the older plugin at which time it can be turned off.

---

**Supported Versions**

The Altiris integration only supports Altiris version 6.5. The integration does not currently Altiris version 7.0.

*Altiris import set data*

This topic lists module names displayed by the Altiris import set data.
The Altiris import set data section shows a list of import set tables used in containing data retrieved from using JDBC to query the Altiris database. The module names (hence the import set tables they point to) match Altiris's table names and structure that it is loading from.

**Module names**

*Note:* Functionality described here requires the Integration - Altiris 2.0 plugin.

- Scheduled Cleanup
  - Configure a schedule to cleanup/delete import set data that have already been transformed
- vComputer
- Inv_AeX_OS_Operating_System
- Inv_AeX_HW_CPU
- Inv_AeX_HW_Memory
- Inv_AeX_HW_Serial_Number
- Inv_AeX_HW_Logical_Disk
- Inv_AeX_OS_Add_Remove_Programs
- Inv_AeX_AC_TCPIP

When viewing each of these table lists, at the end of the list you have links to other operational functions of the import set.

*Figure 483: Altiris integration application*

- **Import Sets**
- **Transform Maps**
- **Transform History** - Log of completed import operations (where an import set was transformed into a table).
- **Edit Web Service**
Web services import set tables for Altiris

This topic will list the modules that define the web service import set tables - the schema for the import set tables that are receiving the JDBC import.

From each web service, you can add/remove fields as well as access the transform maps to make modifications.

**Note:** Functionality described here requires the Integration - Altiris 2.0 plugin.

![Image of Edit Web Service form]

**Related Links**
- Import Sets
- Input Rows
- Transform History

![Image of Import set tables]

Figure 484: Import set tables
Google Maps integration

Map pages enable you to graphically display data on a Google map page based on location data. Maps can be generated using basic JavaScript, but are flexible enough to display even the most complicated of queries. The maps you generate use standard Google Maps API mapping features, including a variety of link types to records in your instance. This feature requires the Google Maps plugin.

Figure 486: Map page

Google Maps plugin

The Google Maps Plugin provides the following configurable properties in System Properties > Google Maps.
## Table 400: Google Maps system properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| google.maps.auto_close | If true, automatically closes a map information window before opening a new one.  
  - Type: true/false  
  - Default value: true  
  - Location: System Properties > Google Maps |
| google.maps.client | Client ID for Google Maps API for Work.  
  - Type: string  
  - Default value: gme-servicenow  
  - Location: System Properties > Google Maps |
| google.maps.private.key | Private key for Google Maps API for Work. This key activates the geolocation feature, which locates users in the system precisely, using data from their mobile devices.  
  - Type: string  
  - Default value: empty  
  - Location: System Properties > Google Maps |
| google.maps.version | Version number of the current installation of Google Maps API.  
  - Type: string  
  - Default value: current version number  
  - Location: System Properties > Google Maps |
| google.maps.key | The Google Maps API key that is tied to the URL of the server. This key authorizes development use of Google Maps API.  
  - Type: string  
  - Default value: empty  
  - Location: System Properties > Google Maps |
| google.maps.latitude | Starting latitude of the map. This value determines the starting position displayed in Google Maps.  
  - Type: string  
  - Default value: 36.008522  
  - Location: System Properties > Google Maps |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.maps.longitude</td>
<td>Starting longitude of the map. This value determines the starting position displayed in Google Maps pages.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: -95.221764</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.max_items</td>
<td>Maximum number of items to display on the map.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 500</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.table</td>
<td>Table used by the map. The table needs the following fields: name, longitude, latitude.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: cmn_location</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
<tr>
<td>google.maps.zoom</td>
<td>Starting zoom level of the map (1 is the lowest)</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: 4</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties &gt; Google Maps</td>
</tr>
</tbody>
</table>

### Google custom search integration

Replace Zing text indexing and search engine with Google Site Search functionality.

Activating custom search integration replaces the default search and provides Google Site Search functionality. You can configure your Google Site Search engine to include or exclude sites in your search results. For more information on the Google Site Search feature, see [Google Site Search](#).

**Activate Google custom search integration**
The Google Custom Search Integration plugin is available by request.

Role required: none

Request the plugin through the HI Service Portal.

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
</tbody>
</table>
Specify the date and time you would like this plugin to be enabled

Date and time must be at least 2 business days from the current time.

**Note:** Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.

<table>
<thead>
<tr>
<th>Reason/Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

**Microsoft SMS/SCCM integration**

The Microsoft SMS/SCCM integration is a one direction import of SMS/SCCM data into ServiceNow’s Configuration Management Database (CMDB).

The integration keeps the ServiceNow CMDB up to date with SMS / SCCM SQLServer database. Relevant data is imported from the SMS / SCCM database to the CMDB. The SMS / SCCM database is considered an authoritative source and is not written to. The import is achieved using a JDBC connection via the MID Server.

This table shows version compatibility between the various Microsoft SMS and SCCM versions and ServiceNow plugins. All versions import identical data, but do it in a different way and with different results in some areas.

**Table 401: Microsoft SCCM version and corresponding ServiceNow Plugin**

<table>
<thead>
<tr>
<th>Supported Microsoft SCCM Version</th>
<th>ServiceNow Integration Plugin</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Microsoft SCCM 2007</td>
<td>Integration - Microsoft SCCM 2007. Also known as the Microsoft SCCM 2007 (3.1.0) plugin.</td>
<td>Use this version when you want to integrate Microsoft SCCM 2007.</td>
</tr>
<tr>
<td>• Microsoft SCCM 2007 R2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Microsoft SCCM 2012 R1</td>
<td>Integration - Microsoft SCCM 2012 v2. Also known as the Microsoft SCCM Integration 4.1.0 plugin.</td>
<td>Use this version when you want to integrate Microsoft SCCM 2012.</td>
</tr>
<tr>
<td>• Microsoft SCCM 2012 R2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Overview of SCCM plugins**

Each Microsoft System Center Configuration Manager (SCCM) plugin has different features. You can use a later version of a SCCM plugin to take advantage of enhanced features.
Table 402: Overview of SCCM plugins

<table>
<thead>
<tr>
<th>Plugin</th>
<th>ServiceNow Plugin Version</th>
<th>Status</th>
<th>Requires SCCM Configuration</th>
<th>Optimized for SCCM 2007</th>
<th>Optimized for SCCM 2012</th>
<th>Asset Intelligence</th>
<th>Incremental Software Reconciliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SCCM 2007</td>
<td>3.1.0</td>
<td>Recommended</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Microsoft SCCM 2012 v2</td>
<td>4.1.0</td>
<td>Recommended</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Asset Intelligence in SCCM

Asset Intelligence is an SCCM feature that can filter, normalize and clean up software records. When enabled, it populates normalized software data in a separate table within the SCCM database. ServiceNow customers may elect to target this separate table instead of the raw software data table. When you import software records into the CMDB with this feature enabled, you get a cleaner set of software data.

Asset Intelligence support is available for Microsoft SCCM 2007 (3.1.0) and Microsoft SCCM 2012v2 (4.1.0). To use this feature, you must enable it in the target ServiceNow instance, as well as in the source SCCM database.

Microsoft SCCM Integration 2007

This version is written based on SCCM 2007, and requires the Integration - Microsoft SCCM 2007 version 3.1.0 plugin. Use this version when you use SCCM 2007.

This version has the following features:

- No flat table configuration is required. You do not need to configure the flat table on the SCCM side.
- Improved performance and an improved process that only collects the changes since the last import for software reconciliation.
- Support for Asset Intelligence.

Software

Only two incremental transforms for software are available by default: one to support software reconciliation without the Software Asset Management plugin and one to support software reconciliation with the Software Asset Management plugin. Full software imports can be invoked by clearing out the Last run datetime field of the SCCM 2007 Software data source.

Microsoft SCCM Integration 2012

The Microsoft System Center Configuration Manager (SCCM) 2012 integration is a one direction import of SCCM data into the ServiceNow Configuration Management Database (CMDB). Scheduled imports bring relevant data from SCCM tables into the CMDB through the SQL database and map them to the ServiceNow instance. Beginning with this version, either a full data import or an incremental import can be performed. The import is achieved using a JDBC connection via the MID Server. This functionality requires the Integration - Microsoft SCCM plugin and a new table in the SQL Server database.

For SCCM 2012, ServiceNow supports the following SCCM versions:
### Table 403: Supported Version

<table>
<thead>
<tr>
<th>SCCM Plugin</th>
<th>Supported Microsoft SCCM Versions</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Microsoft SCCM 2012R2 server has not been tested by ServiceNow. However, customers have reported success using it with the Integration - SCCM 2012 plugin</td>
</tr>
</tbody>
</table>

### Version 2012 Overview

Version 2012 is designed to work with SCCM 2012 release 1.

It comes with default table views and related column names from SCCM 2012. Version 2012 can work with SCCM 2007, but it will require configurations on the table name and column names for certain data source imports.

A major difference between version 3.0 and SCCM 2012 is that a flat table is no longer needed for the import. Instead, data is imported from the Computer Identity data source table within SCCM. The table can be easily extended and configured to include additional information.

SCCM 2012 also provides three transform maps for a software import. Only one transform map can be enabled at a time.

1. SCCM 2012 Software - Full Import
2. Incremental Import
3. Incremental Import (SAM enabled)

The first transform map performs a full import of the software data. This transform map automatically populates into the cmdb_ci_spkg and cmdb_software_instance tables or the cmdb_sam_sw_install table depending on if the Software Asset Management plugin is enabled. If the plugin is not enabled, the former tables are populated; if the plugin is enabled, the latter table is populated.

The second and third transform maps perform an incremental import and should be enabled or disabled depending on whether the Software Asset Management plugin was enabled. If the plugin is not enabled, use "Incremental import"; otherwise, use "Incremental import (SAM enabled)". By default, the incremental import transform map is enabled.

**Note:** ServiceNow SCCM Integration version 2012 supports Microsoft SCCM 2012.

### Identifiers

The Microsoft SCCM integration uses CI identification to update CIs created from data imported from SCCM with a resource ID. A Discovery Identifier called SCCM ID & Class Name returns the resource ID of a computer from SCCM and stores it in a table called Source [sys_object_source]. When resource IDs are first imported, either from SCCM or Discovery, ServiceNow populates the Source table with IDs for each CI it identifies. In subsequent imports, if an incoming ID matches that of an existing CI, ServiceNow
updates the information for that CI in the CMDB. If the incoming resource ID does not match that of an existing CI, ServiceNow creates a new CI and populates it with the resource ID.

**Scripts**

Data population scripts added with the Integration - Microsoft SCCM plugin populate the related data in the CMDB for each target CI discovered by the SCCM ID & Class Name identifier.

**Software**

The Microsoft SCCM Integration version 2012 reconciles the software package count in the records for a CI and removes a software instance from the Software Instance `[cmdb_software_instance]` table if the software package is uninstalled from the CI.

**Note:** The table data imported from SCCM in Microsoft SCCM Integration version 2012 must contain complete data for the CI. ServiceNow assumes that the import represents all relationships that exist and adjusts the CMDB accordingly. Partial data received from SCCM tables can cause the deletion of active relationships.

For a full import, the resulting relationship looks like this:

![Figure 487: SCCM 2012 Full Diagram](image)

For an incremental import, the resulting relationship looks like this:
Perform Full Import for SCCM 2007 or SCCM 2012
You must re-import all the software records when switching to the SCCM 2007 or SCCM 2012v2 plugin. Only a full import from Microsoft SCCM will allow correct data reconciliation.

Both plugins track the incremental import of information about software that was uninstalled from CIs. To support this feature, the following fields where added to the tables cmdb_software_instance and cmdb_sam_sw_install (when Software Asset Management plugin is enabled):

- The SCCM timestamp [sccm_timestamp] field.
- The SCCM group ID [sccm_group_id] field.

Note: Unlike previous SCCM integration plugins, the same transform map is used for full imports and incremental imports in SCCM 2007 and SCCM 2012v2.

To perform a full data import:
1. Navigate to Integration - Microsoft SCCM 20xx > Data Sources.
2. Click SCCM 20xx Software.
3. Clear the Use last run date time box.
4. Click Update.
5. Navigate to Integration - Microsoft SCCM 20xx > Scheduled Import.
6. Click Execute Now.

Verizon eBonding integration

This topic outlines the tasks required for moving the Verizon eBonding integration from a Dev/Test environment to a Production environment.

This process requires coordination with your assigned Verizon implementation manager.
General Tasks

The following work can be performed any time:

- Obtain a production SOAP password from Verizon. Your Verizon SOAP user ID is the same as that used against the Verizon test broker.
- Create an integration user for Verizon in the ServiceNow Production instance. Use the same user name and password that was created for the development instance, so that the communication between Verizon and ServiceNow is properly authenticated.

Migrating to a Production Environment

Perform these tasks in order.

- Task 1: Request the Verizon eBonding plugin for the production instance. When the plugin is installed, the only immediate user interface change is an additional **Category** field named **VZ eBonding**. All of the triggers for the integration are dependent on the category being set to **VZ eBonding**. Enabling the plugin on the production instance should not affect your users. If you wish to take additional precautions, disable the **VZ eBonding** category from the list when the plugin is activated for the instance.
- Task 2: Set up the Verizon certificate and keystore. Copy these directly over from the Dev instance or regenerate them so that you have separate versions for the Dev and Production instances.
- Task 3: Configure the integration properties to work with the Production Verizon eBonding system. This is userID, password, and other properties provided by your Verizon representative.
- Task 4: Change all the Verizon eBroker SOAP Message functions to use Verizon’s production SOAP Endpoint. In most cases, the endpoint URL will be: https://pubwebsvc.vzbi.com:443/Trouble_Management_v3r0. Verify this URL with the Verizon implementation specialist.
  - In the **SOAP Message Functions** related list, click each function, clear the **Lock** check box, and enter the production endpoint.
  - Configure the **SOAP Message Functions** related list and add the **SOAP endpoint** column. In the **SOAP Message Functions** list view, unlock and update the endpoint for all records by editing multiple records.
- Task 5: Apply any customizations. If customizations to the integration on the Dev system were captured in an update set, apply that update set to the Production system.
- Task 6: Provide Verizon with the production SOAP Endpoint. The SOAP endpoint should look something like this: https://yourproductioninstance.service-now.com/ETMSPublish.do?WSDL.
- Task 7: Coordinate a Production turn up with your Verizon representative to do basic testing.

**Warning:** Do not create test tickets against the production system without working with the Verizon team. Any tickets created are treated as real tickets.

Web services

This section will list the modules that define the web service import set tables - the schema for the import set tables that are receiving the JDBC import.

From each web service, you can add/remove fields, as well as access the transform maps to make modifications.

CMDB import integrations

A class of integrations with existing configuration management databases (CMDB) that imports CIs into ServiceNow CMDB.
CMDB import integrations use a combination of Import Sets, JDBC data sources, and a MID server \((JDBCProbe)\). Example integrations that fall into this category include:

- Altiris
- Microsoft SCCM / SMS

![Diagram of CMDB import process]

**Figure 489: CMDB Import**

**Import set data**

This section of the application lists the import set tables that have been predefined for the external CMDB we are integrating with.

Selecting the tables will display the data already retrieved from a JDBC data source import, including the import set it is associated with, its state, as well as any information related to the transformation of the import set table data row.

**Configuration**

After enabling the plugin for an integration of this type, you will have a new application that consists of the following common modules. Next, install a MID Server in your environment that will be used to execute JDBC queries connecting with your database. This MID server will require port level access to your database. For example, SQL Server port 1433.
Figure 490: CMDB Import JDBC Integration

Table 404: CMDB Import JDBC Integration

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup</td>
<td>• Configure the data sources from one form</td>
</tr>
<tr>
<td></td>
<td>• Specify Database server settings and MID server</td>
</tr>
<tr>
<td></td>
<td>• Test configuration</td>
</tr>
<tr>
<td>Scheduled Import</td>
<td>Schedule the execution of the import or import immediately</td>
</tr>
<tr>
<td>Data Sources</td>
<td>A list of the pre-configured data sources defining the external CMDB database</td>
</tr>
<tr>
<td>Progress</td>
<td>A historical list of progress on scheduled imports</td>
</tr>
<tr>
<td>Transform History</td>
<td>A historical list of transformations performed during scheduled imports</td>
</tr>
</tbody>
</table>

**Setup**

The Setup module allows the administrator to configure the JDBC data sources for the external CMDB tables. When you save the form, it will apply the changes to all data sources under this integration.
Figure 491: Integration Setup form

Table 405: Integration Setup form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Server</td>
<td>The database server host name of IP address, if executed via a MID server you have installed, this server needs to be accessible via your MID server</td>
</tr>
<tr>
<td>Database Name</td>
<td>The database name for your CMDB</td>
</tr>
<tr>
<td>Database User ID</td>
<td>The user ID used to connect to your CMDB database, it must have sufficient privileges to access data defined by your data sources. ServiceNow only supports SQL account credentials, you cannot authenticate using a domain user.</td>
</tr>
<tr>
<td>Database User Password</td>
<td>The user password for the database user ID</td>
</tr>
<tr>
<td>MID Server</td>
<td>The MID server to use for executing the database query on. You will want to install your own MID server behind your firewall if your ServiceNow instance does not have a direct JDBC connection to your database server</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the MID server: Up or Down</td>
</tr>
</tbody>
</table>
Test Data Source Connections

After your setup values are inputted and saved, a **Test data source connections** UI action is available. Executing this action will perform a database table row count on each table and query configured in each data source. The following image shows a successful test.

![Progress Table]

<table>
<thead>
<tr>
<th>Name</th>
<th>Testing JDBCProbes</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Complete</td>
</tr>
<tr>
<td>Completion code</td>
<td>Success</td>
</tr>
<tr>
<td>Message</td>
<td></td>
</tr>
<tr>
<td>Test results:</td>
<td></td>
</tr>
</tbody>
</table>

```
SELECT count(*) as count FROM System_DATA
Result: 2312 (582 ms)

SELECT count(*) as count FROM Processor_DATA
Result: 3428 (748 ms)

SELECT count(*) as count FROM Add_Remove_Programs_DATA
Result: 232154 (619 ms)

SELECT count(*) as count FROM Computer_System_DATA
Result: 2312 (501 ms)

SELECT count(*) as count FROM Operating_System_DATA
Result: 2312 (412 ms)

SELECT count(*) as count FROM PC_BIOS_DATA
Result: 2312 (1136 ms)
```

**Next steps...**

- **Setup** Return to the setup form
- **Scheduled Import** Proceed to setup schedule for import

Figure 492: Successful Test
MID server integrations

The system provides integrations that connect through a MID Server.

Syslog probe

The ServiceNow Syslog probe uses the MID Server to deliver log messages from a ServiceNow instance to another machine, such as a dedicated log server, using the syslog protocol over an IP network.

How the Syslog Probe Works

The syslog probe is launched by a ServiceNow Script Include (called Syslog) that can be invoked from a business rule, event, or Orchestration activity and is launched by a MID Server. A syslog server or any server that can receive messages using the syslog protocol must be installed on the recipient (target) machine. Typically, a dedicated log server in the network is configured to receive all internal syslog messages. Some products that accept syslog messages are:

- ArcSight
- Splunk
- LogLogic
- syslog-ng

Example

The Acme Corporation wants to send a log message from their ServiceNow instance to an ArcSight syslog server inside their corporate firewall each time a user login fails. The system administrator uses the login.failed event to trigger a business rule that invokes the Syslog Script Include each time a login fails. Acme’s MID Server checks the ECC Queue for work and picks up the syslog probe, which contains the log entry. The MID Server then sends the log message to the ArcSight server, which gathers log messages from all the machine in the internal network.
Figure 493: Syslog Probe Diagram

Code Sample

The following code sample, included in an event or a business rule, directly calls the Syslog Script Include and sends a syslog message to a designated syslog server:

```javascript
var sl = new Syslog('syslog.service-now.com ', 'mid.server.Eclipse', 16);
sl.log('This is a sample log message', 6);
```

This code does the following:
- Sends the log message to facility 16
• Sets the priority at 6 (informational)
• Sends the message to the syslog.service-now.com syslog server
• Launches the probe via the Eclipse MID Server

**Supported integration interfaces**

ServiceNow provides a number of interfaces to be able to directly integrate with the platform. These interfaces are considered part of the platform and are provided at no additional charge.

<table>
<thead>
<tr>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
</tr>
<tr>
<td>JDBC</td>
</tr>
<tr>
<td>JSON</td>
</tr>
<tr>
<td>LDAP</td>
</tr>
<tr>
<td>SOAP</td>
</tr>
<tr>
<td>REST</td>
</tr>
<tr>
<td>SSO - SAML 2.0</td>
</tr>
<tr>
<td>Digest Authentication</td>
</tr>
<tr>
<td>ODBC</td>
</tr>
<tr>
<td>Data Export</td>
</tr>
<tr>
<td>CTI</td>
</tr>
</tbody>
</table>

**Email Integrations**

The system provides integrations through email.

*Change Management Integration with Outlook*

ServiceNow's Change Management application can send your pending changes to your Microsoft Outlook Calendar automatically. This functionality is controlled by a script in the change events business rule.

**Note:** The following integration is not part of the supported platform. The integration may require additional customization prior to operating correctly. This integration is typically performed by an experienced administrator or consultant.

Outlook uses the standard iCalendar data format, which is simply a standardized way to describe a calendar event. Events in this format can be sent from a meeting originator (ServiceNow) to a meeting attendee (that's you) as specially formatted emails. So when a change request is assigned to you, ServiceNow will automatically send you an iCalendar-formatted email. You'll see the meeting request inside of Outlook, and you can click accept to add it to your calendar.
If you've turned on Calendar Integration, you'll be informed every time:

- You're assigned to a schedulable change
- A schedulable change of yours gets reassigned to somebody else
- A change gets rescheduled for a different time

**Note:** Note the deliberate use of the phrase schedulable change. A change needs a start date and an end date before it can be scheduled properly, otherwise it isn't considered schedulable and therefore won't show up in Outlook.

**Common Questions and Answers**

- If I change the start or end of a change in Outlook, will ServiceNow be updated?
  
  No, it won't. Currently this is a one-way integration. ServiceNow can send scheduled events to Outlook, but the reverse is not true.

- Does this integration work with other email clients?
  
  Yes. While we've only tested with Outlook 2003 and 2007, this integration should work with any email client that supports the iCalendar format.

- What about older versions of Outlook like Outlook 2000?
  
  There are known issues with Microsoft's Outlook 2000 iCalendar support. Many of these issues (though not all) can be resolved by installing Microsoft patches.

**Change Management Business Rule**

The following is the "change events" business rule associated with this enhancement:

```java
if (current.start_date.changes() || current.end_date.changes() || current.assigned_to.changes()) {
  if (!current.start_date.nil() && !current.end_date.nil() && !current.assigned_to.nil()) {
    gs.eventQueue("change.calendar.notify", current, current.assigned_to, previous.assigned_to);
  }
}
```
// Remove from previous assigned to, due to assigned_to changing
if (!previous.assigned_to.nil()) {
    if (!current.assigned_to.nil() && current.assigned_to.changes() &&
        (!previous.start_date.nil() && !previous.end_date.nil())) {
        gs.eventQueue("change.calendar.notify.remove", current,
                      current.assigned_to, previous.assigned_to);
    }
}

// Remove old calendar from current assigned to, due to date changing
else if (!current.assigned_to.nil()) {
    if ((current.start_date.changes() && !previous.start_date.nil()) ||
        (current.end_date.changes() && !previous.end_date.nil())) {
        gs.eventQueue("change.calendar.notify.remove", current,
                      current.assigned_to, current.assigned_to);
    }
}

Turn on Change Management Integration with Outlook
You can decide on a per user basis whether change requests assigned to that user are sent to their
Outlook calendar.

Naturally, if they don't use Outlook, you might not want to turn this option on. Even if they do have Outlook,
they may prefer to not receive calendar updates.

To turn on Outlook integration:
1. Open the user record for the user for whom you want to enable this feature.
2. Change the Calendar Integration field from None to Outlook.

   Tip: You may need to add this field to your User Record by configuring the form.

3. Save the record.
   Any changes assigned to that user generates calendar integration messages via email.

Event management integration

The purpose of an event management integration is to collect event-based information and convert it into
tasks or tickets within ServiceNow. When updates occur, the integration sends changes back to event
originator.

The integration collects source event data from an inbound web service invocation and transforms it into
target data as a ServiceNow ITIL service object. Data coalescing helps filter many source records into one
target record.
When acting as an event consumer, ServiceNow only accept SOAP inserts. ServiceNow uses a Web Service Import Set (WSIS) for event notification. The web service uses a synchronous transform, and the transform responds with a target detail record.

**Event notification**
Event management integrations monitor ITIL service objects. When ServiceNow updates or creates a monitored service object, the integration sends an event notification to the external service and identifies the service object by a common key.

**Figure 496: Event notification**
- Service objects have a unique Identifier on the event generator and event consumer systems.
- When using web services for communications, ServiceNow business rules or listeners send SOAP messages.
  - SOAPEnvelope
  - SOAPRequest
• When using a MID Server for communications, ServiceNow business rules or listeners queue work for the MID Server
  • The MID Server updates or adds service objects on the event consumer using CLI, Java API or JDBC

Computer Telephony Integration

A Computer Telephony Integration (CTI) is accomplished in ServiceNow simply by the external CTI client on the user machines formatting a URL to the ServiceNow instance.

The URL must have the following components:

1. The base URL. For example: https://<instance name>.service-now.com/cti.do? would get to the instance and ask for CTI processing
2. Parameters identify what parts of the incident form to display.
   • sysparm_caller_name=name where 'name' is the name for a user within the SNC system
   • sysparm_caller_phone=phone where 'phone' is the the phone number for a user within the SNC System. Either name or phone should be provided if you want to identify the user on the call. Other parameters may be supplied to identify the user as discussed later.
   • sysparm_task_id=taskID where 'taskID' identifies an existing issue within the system that the caller is calling about.
   • sysparm_view=view where 'view' is the name of the view within the SNC system that should be used to display the data
   • sysparm_xxxx=value where 'xxxx' is the name of a field within the 'incident' record that should be populated with the specified 'value'. For example sysparm_priority=1 would result in the priority field set to value of 1 when the new incident screen is popped.
   • sysparm_cti_rule=name is the name of a business rule that should be invoked for CTI processing rather than using the default behavior.

Example

So an example URL to bring up a screen shot for user Don Goodliffe would look like the following
https://<instance name>.service-now.com/cti.do?sysparm_caller_name=Don%20Goodliffe

The default activity that will happen in SNC when getting a CTI request is the following:

1. The system will try to identify the user by the sysparm_caller_name value if it was supplied
2. If no user has been found, the system will try to identify the user by the sysparm_caller_phone value if it was supplied
3. If a user has been identified then one of the following will occur
   a. If the user has incidents that are currently open the screen pop will be the caller screen pop that shows information about the current caller and all the incidents the caller has currently opened
   b. If the user does not have any open incidents, the screen pop will be for a new incident will all information provided in the cti URL filled in.
4. If a user was not identified and a taskID was given and the taskID exists, then the screen pop will be to show the details for the task.

If the default behavior is not what is wanted then the installation can write a business rule that decides what should be done. The business rule function name has a default value of 'cti' but can be overridden with a sysparm_cti_rule value in the CTI URL.
The business rule must return the URL that should be used for the screen pop. A sample business rule is supplied in the system with a name of “CTI Processing”. This business rule actually implements the ‘default’ behavior as described above.

Any parameter that was on the CTI URL is available to the business rule as a global value. For example to get the value of the caller name:

```
var name = sysparm_caller_name
```

Multiple sysparm parameters can be used, separated by ampersands (&).

Here is a screenshot of the result of calling the above mentioned URL:

![Figure 497: Computer Telephony Integration](image.png)
Linking to the ServiceNow login page

There are several ways you can add a ServiceNow login link to your intranet.

You can add a login link by:

- Enabling the PortletLogin Script include to be Client Callable.
- Creating a simple HTML link to your instance that takes your users directly to the ServiceNow login page.
- Adding an iframe link to the ServiceNow login portlet in one of your HTML pages to permit direct login.

Creating a Simple Link

Edit a web page on your intranet and add a direct link to your ServiceNow instance.

For example:

```html
<a href="https://yourinstance.service-now.com">Help Desk</a>
```

Enabling the PortletLogin Script include to be Client Callable

The login portlet (https://yourinstance.service-now.com/portal/login.html) uses the PortletLogin Script Include.

Since this script is not client callable by default, you must enable the client callable option.

1. Navigate to **System UI > Script Includes**.
2. Open the script **PortletLogin**.
3. Check the **Client Callable** check box.
4. Click **Update**.

Add the Login Portlet

Adding the login portlet to an iframe HTML element creates an unbranded user and password prompt on any HTML page.

The user enters their user id and password and either presses the ENTER key or clicks the Login button. Once validated, the browser transfers to the logged-in ServiceNow session.

The portlet's size is 240 pixels wide and 125 pixels high. The portlet looks like:

```
User:  
Password:  
Login
```

![Figure 498: Portal Login](image)

To enable the login portlet:

1. Set the high security properties `glide.security.use_csrf_token` and `glide.set_x_frame_options` to **false**.
2. Add the following HTML code to your portal:

```html
<iframe border="0" frameborder="0" width="240" height="125" src="https://yourinstance.service-now.com/portal/login.htm"></iframe>
```

If you do not want to change your instance's High Security settings, create a simple link instead.

© 2017 ServiceNow. All rights reserved. 1713
JDBCProbe

A JDBC probe runs on the MID Server to query an external database via [JDBC] and returns results to ServiceNowServiceNow.

Probes interact with the MID Server via the ECC Queue, therefore the response of a JDBC probe returns as an XML payload in an "input" ECC Queue record. By default, each response payload will contain up to 200 returned rows, this value can be modified by setting the probe parameter jdbcprobe_result_set_rows to the desired number.

Activating the Plugin

Contact Customer Support to activate the Integration - JDBC (com.snc.integration.jdbc) plugin.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   If the plugin depends on other plugins, these plugins and their activation status are listed.
3. [Optional] Select the Load demo data check box.
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when first activating the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Direct JDBC Probe

A direct JDBC probe specifies all the parameters necessary in the outbound ECC Queue XML payload.

It is a standalone probe that is decoupled from a JDBC Data Source and does not insert into an import set. The required field in the ECC Queue record for a direct JDBC probe is Topic and it must equal JDBCProbe. The Source field is reserved for the sys_id of the data source record in the case of a JDBC data source.

Note: If you have ServiceNowServiceNow Discovery enabled on your platform, add the skip_sensor parameter to the probe to avoid the discovery sensors from processing the probe and resulting in an "No sensors defined" error.

XML Structure

A direct JDBC probe has the following XML payload structure

```xml
<parameters>
  <parameter name="skip_sensor" value="true"/>
  <parameter name="parameter_name" value="parameter_value"/>
  ...
  <parameter name="work"><select>...</select></parameter>
</parameters>
```

JDBC Probes via Data Source

JDBC probes are executed via a JDBC data source when an import is running against the data source.

A JDBC data source JDBC probe is described by the JDBCProbe Topic and the sys_id of the data source in the Source field of the ECC Queue output record.

The data source record would look like this
The following ECC Queue output probe will be created when you *load* from the data source.
**Figure 500: JDBC Data Source Probe**

*Select * JDBC Probe short cut

Alternatively, you may specify a table_name parameter instead of a work element and the following query could be executed.

```sql
select * from <table_name>
```

For example, using the following XML payload

```xml
<?xml version= "1.0" encoding= "UTF-8" ?><parameters><parameter name = "jdbc_driver" value = "com.microsoft.sqlserver.jdbc.SQLServerDriver" />
<parameter name = "connection_string" value = "jdbc:sqlserver://xxx.service-now.com;databaseName=SMS_CRICKET;user=sms;password=sms" />
<parameter name = "table_name" value = "System_DATA" /></parameters>
```

**Counting Rows**

To count the number of rows in a table, you can indicate a select count(*) type query by including the count_rows parameter with a value of true.

For example

```xml
<?xml version= "1.0" encoding= "UTF-8" ?><parameters><parameter name = "jdbc_driver" value = "com.microsoft.sqlserver.jdbc.SQLServerDriver" />
<parameter name = "connection_string" value = "jdbc:sqlserver://xxx.service-now.com;databaseName=SMS;user=sms;password=sms" />
<parameter name = "count_rows" value = "true" />
<parameter name = "table_name" value = "System_DATA" /></parameters>
```

The resulting response XML payload

```xml
<parameters>
...
<result query = "SELECT count(*) as row_count FROM System_DATA" ><row id = "1" ><row_count class = "java.lang.Integer" length = "11" type = "4" >2312</row_count></row></result>
```
Parameters
The following parameters are available in a direct JDBC probe.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Example</th>
<th>Required?</th>
</tr>
</thead>
</table>
| jdbc_driver    | The Java class name for the JDBC driver to use, the currently supported drivers are:  
  • Oracle - oracle.jdbc.OracleDriver  
  • Microsoft SQL Server - com.microsoft.sqlserver.jdbc.SQLServerDriver  
  • MySQL - com.mysql.jdbc.Driver | com.microsoft.sqlserver.jdbc.SQLServerDriver | Y         |
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Example</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection_string</td>
<td>The JDBC connection string/URL for defining the connection, usually contains information about the database server and name, the user id and password for connecting to the database. The syntax is vendor specific, refer to the following links for reference</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Oracle - [jdbc:oracle:thin:&lt;username/password&gt;@&lt;database&gt;]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Microsoft SQL Server - [jdbc:sqlserver://localhost;user=MyUserName;password=*****;]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• MySQL - [jdbc:mysql://localhost/database?user=username%26password=passwd]</td>
<td>jdbc:sqlserver://xxx.service-now.com;databaseName=SMS;user=sms_user;password=sms_password</td>
<td>Y</td>
</tr>
</tbody>
</table>
| work | A parent element of an XML fragment describing the SQL command to execute | ...<parameter name= "work" >
  <select table = "System_DATA" where = "InstanceKey=692"
        >
      <MachineID />
      <SMSID0 />
    </select>
  </parameter>...
<p>| Y (if not using the table_name parameter short cut) |
| query_timeout | Specify the number of seconds the JDBC driver will wait for a query (SELECT) to complete. Zero means no timeout. If timeout is exceeded, the integration considers the JDBC result inaccessible and places it in an error state. | 60 | N (optional during SELECT) |</p>
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Example</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>table_name</td>
<td>An alternate way of executing a <code>select *</code> query, instead of using the <code>work</code> element. Equivalent to executing &quot;select * from &lt;table_name&gt;&quot;</td>
<td>System_DATA</td>
<td>N (required during count_rows == true)</td>
</tr>
<tr>
<td>count_rows</td>
<td>Indicate the a <code>select count(*)</code> query should be executed returning the number of rows. The result of the count will be returned as a <code>row_count</code> element in the <code>result</code> element.</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>query</td>
<td>Indicate the type of query. If the query type is &quot;Specific SQL&quot;, the sql_statement will be required to specify the SQL statement.</td>
<td>The possible choices are &quot;All Rows from Table&quot; or &quot;Specific SQL&quot;.</td>
<td>N</td>
</tr>
<tr>
<td>sql_statement</td>
<td>Use a specific SQL query. The presence of this element will execute a direct query specified in the value attribute.</td>
<td>select * from any_table where id = 123</td>
<td>N</td>
</tr>
<tr>
<td>skip_sensor</td>
<td>Determines if Discovery will attempt to process the ECC input from the JDBCProbe. This parameter defaults to true.</td>
<td>true</td>
<td>N</td>
</tr>
</tbody>
</table>

For example, to query a table using a direct JDBC probe, requires the following parameters:

- JDBC driver class name
- JDBC connection string
  - Database server
  - Database name
  - User name
  - User password
- The table name
Figure 501: Queue form

*Using the Work Element*

The work element encodes SQL statements to be executed by the probe.

The following are the valid `work` child elements.

- `select`
- `update`
- `insert`
- `delete`

**SELECT**

Retrieve rows from the specified table in the database specified by the JDBC connection string. The simplest `select` work just specifies the table name and will retrieve all fields in a row, for example:

```xml

...<parameter name= "work" ><select table = "System_DATA" /></parameter>...

```

To specify a search criteria, specify the `where` attribute on the `select` element, for example:

```xml

... ...

```
To specify the fields you want returned, embed the fields as child elements of the `select` element, for example:

```xml
<parameter name="work" ><select table = "System_DATA" where = "InstanceKey=692" /></parameter>
```

To use a direct SQL statement, specify it in the `query` parameter

```xml
<parameter name="query" value="Specific SQL" /><parameter name="sql_statement" value = "select * from any_table where value='test'" />
```

**UPDATE**

An `update` SQL can be executed by specifying the `table`, optional `where` clause and required child elements for the fields and their values to update.

```xml
<parameter name="work" ><update table = "alerts.status" where = "ServerName %= 'NCOMS' AND ServerSerial=3935" ><Agent>ServiceNow - INC10020</Agent><URL>http://Macintosh-9.local:8080/glide/incident.do?sys_id=17a31f380a0a0bae0048ca875c8891d0</URL><Severity quoted = "false" >3</Severity><Acknowledged quoted = "false" >0</Acknowledged></update></parameter>
```

**INSERT**

An `insert` SQL can be executed by specifying the `table`, and required child elements for the fields and their values to insert.

```xml
<parameter name="work" ><insert table = "alerts.status" ><Agent>ServiceNow - INC10020</Agent><URL>http://Macintosh-9.local:8080/glide/incident.do?sys_id=17a31f380a0a0bae0048ca875c8891d0</URL><Severity quoted = "false" >3</Severity><Acknowledged quoted = "false" >0</Acknowledged></insert></parameter>
```

**DELETE**

A `delete` SQL can be executed by specifying the `table`, and optional `where` clause.

```xml
<parameter name="work" ><delete table = "alerts.status" where = "ServerName %= 'NCOMS' AND ServerSerial=3935" /></parameter>
```
User administration

Manage the individuals who can access ServiceNow by defining them as users in the system and assigning them to groups. Use the session control options to terminate ServiceNow sessions, for example when system maintenance is required. Create roles that provide selective access to ServiceNow functionality, then assign the roles to groups when all associated users need to access that functionality, or to individual users.

Administrative roles

In previous versions of the product, a great many administrative tasks and rights were granted through the admin role. The rights to change business rules, client scripts, and UI policy, and to create script includes were all controlled by this role. In the current system, a family of more granular administrative roles allows the granting of more specific rights to individual users without granting the broader privileges of the admin role.

For example, an administrator can now grant a user the rights to change UI policy, but not the rights to edit client scripts.

Note: The addition of these new roles does not change the existing behavior of the admin role, which still retains its general administrative privileges on the platform.

Roles

Roles control access to features and capabilities in applications and modules.

After access has been granted to a role, all of the groups or users assigned to the role are granted the access. Roles can contain other roles, and any access granted to a role is granted to any role that contains it.

For a complete list of the roles included with the ServiceNow platform, see Base system roles on page 1724.

Create a role

Create a role to control access to features and capabilities in applications and modules.

Once access has been granted to a role, all of the groups or users assigned to the role are granted the access. Roles can contain other roles, and any access granted to a role is granted to any role that contains it.

For a complete list of the roles included with the base instance, see Base System Roles.

1. Navigate to User Administration > Roles.
2. Click New.
3. Fill in the fields on the form and then click Submit.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the role.</td>
</tr>
<tr>
<td>Application</td>
<td>Select the application that contains this record.</td>
</tr>
</tbody>
</table>
### Field
<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevated privilege</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

The new role appears on the Roles list. The new role does not have access to any application or module until you add other roles to it or add the new role to the appropriate applications and modules.

**Add a role to an existing role**

When you add a new role to an existing role for a user, the user inherits the access that is granted by the new role.

1. Open the existing role and click **Edit** in the **Contains Roles** related list.
2. Use the slushbucket to add one or more roles to the existing role.
3. Click **Save**.

The users with the existing role inherit the access that is granted by the new role.

**Assign a role to a group**

You can assign a role to a group to grant access to applications and modules to group members.

Role required: user_admin or admin

When you assign roles to groups rather than to individual users, members of the group inherit the role. When a user switches groups, the new group role is assigned automatically.

1. Navigate to **User Administration** > **Groups**.
2. Click the group to assign a role.
3. In the **Roles** related list, click **Edit**.
4. Use the slushbucket to add the desired roles to the group.
5. Click **Save**.

**Create a group role**

Create a group role to control access to features and capabilities in applications for all members in a group.

1. Navigate to **User Administration** > **Group Roles**.
2. Click **New**.
3. Fill in the fields on the form and then click **Submit**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>[Mandatory] Select a group.</td>
</tr>
<tr>
<td>Role</td>
<td>[Mandatory] Select the role to apply to the group.</td>
</tr>
</tbody>
</table>
Grant a role access to applications and modules

Roles control access to features and capabilities in applications and modules. You add a role to an application or module to enable the role to grant access to the application or module for all users with the role.

1. Navigate to **System Definition > Applications** or **System Definition > Modules**.
2. Click the appropriate application or module to open it in form view.
3. Click the lock to open the **Roles** field.
4. Use the slushbucket to add the desired roles to the application or module.
5. Click the lock to close the **Roles** field, and then save your changes.

Base system roles

Administrators can assign one or more base system user roles to grant access to base system platform features and applications.

The following standard roles are included in the base ServiceNow system with a new instance.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>The administrator role. This role has special access to all system features, functions, and data because administrators can override ACL rules and pass all role checks. Consider these implications when using admin overrides on ACLs. If you have sensitive information, such as HR records, that you need to protect, you must create a custom admin role for that area and train a person authorized to see those records to act as the administrator. Also note the <em>Special Administrative Roles</em>.</td>
</tr>
<tr>
<td>agent_admin</td>
<td>Can manage MID Server-related scripts.</td>
</tr>
<tr>
<td>approval_admin</td>
<td>Can approve or reject approvals.</td>
</tr>
</tbody>
</table>

*Note:* The system does not support changing the name of any base system role. Changing the name of a base system role will prevent users and groups from accessing base system resources that depend on these roles.
<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>approver_user</td>
<td>Can modify requests for approval routed to them. They also have all capabilities of Requesters.</td>
</tr>
<tr>
<td>assign_rule_admin</td>
<td>Can manage Assignment Rules.</td>
</tr>
<tr>
<td>asset</td>
<td>Can manage hardware and software assets.</td>
</tr>
<tr>
<td>catalog</td>
<td>Has access to service catalog requests.</td>
</tr>
<tr>
<td>catalog_admin</td>
<td>Can manage the Service Catalog application, including catalog categories and items.</td>
</tr>
<tr>
<td>catalog_editor</td>
<td>Can create, modify, and publish items within categories they are assigned to.</td>
</tr>
<tr>
<td>catalog_item_designer</td>
<td>Can view the status of their category requests.</td>
</tr>
<tr>
<td>catalog_manager</td>
<td>Can view and assign catalog editors to their categories. Can also create, modify, and publish items within their categories.</td>
</tr>
<tr>
<td>category_manager</td>
<td>Can create, edit, and delete model categories.</td>
</tr>
<tr>
<td>contract_manager</td>
<td>Can create, edit, and delete contracts through the Contract Management application.</td>
</tr>
<tr>
<td>ecmdb_admin</td>
<td>Can administer the CMDB.</td>
</tr>
<tr>
<td>filter_admin</td>
<td>Can manage filters.</td>
</tr>
<tr>
<td>filter_global</td>
<td>Can create global filters.</td>
</tr>
<tr>
<td>filter_group</td>
<td>Can create filters that belong to groups of which the user is a member.</td>
</tr>
<tr>
<td>gauge_maker</td>
<td>Can create gauges from reports or charts.</td>
</tr>
<tr>
<td>image_admin</td>
<td>Can manage image files on the Images [db_image] table.</td>
</tr>
<tr>
<td>impersonator</td>
<td>Can impersonate users. Does not allow impersonation of admin users.</td>
</tr>
<tr>
<td>import_admin</td>
<td>Can manage all aspects of import sets and imports.</td>
</tr>
<tr>
<td>import_scheduler</td>
<td>Can schedule imports.</td>
</tr>
<tr>
<td>import_set_loader</td>
<td>Can load import sets.</td>
</tr>
<tr>
<td>import_transformer</td>
<td>Can manage import set transform maps and run transforms.</td>
</tr>
<tr>
<td>inventory_admin</td>
<td>Can create and delete stock information. Only users with the inventory_admin role can edit stock rules, stockrooms, and stockroom types.</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>inventory_user</td>
<td>Has access to stock information. Can create and manage transfer orders.</td>
</tr>
<tr>
<td>itil</td>
<td>Can perform standard actions for an ITIL helpdesk technician. Can open, update, close incidents, problems, changes, configuration management items. By default, only users with the itil role can have tasks assigned to them.</td>
</tr>
<tr>
<td>itil_admin</td>
<td>Possesses more privileges than the itil role and is intended for team leads. This role has the ability to delete incidents, problems, changes, and other related entities when both the itil and itil_admin roles are assigned.</td>
</tr>
<tr>
<td>knowledge</td>
<td>Can create, edit, and review knowledge base articles.</td>
</tr>
<tr>
<td>knowledge_admin</td>
<td>Can manage the knowledge base.</td>
</tr>
<tr>
<td>list_updater</td>
<td>Can use Update Entire List and Update Selected menu options on lists.</td>
</tr>
<tr>
<td>maint</td>
<td>Reserved for ServiceNow use.</td>
</tr>
<tr>
<td>mid_server</td>
<td>Role that any MID server user should be granted. This role gives the MID server access to the tables it ordinarily uses.</td>
</tr>
<tr>
<td>model_manager</td>
<td>Can create new CMDB models. Model manager can control the base models and any model extensions that are not hardware, software, or consumables. Hardware and consumable models are controlled by the asset manager role (asset). Software models are control by the software asset manager role (sam).</td>
</tr>
<tr>
<td>nobody</td>
<td>The nobody role means that nobody has access - not even admin or maint. The nobody role takes precedence over the admin override option on ACLs, so even admins cannot have access. See Create an ACL rule on page 2515.</td>
</tr>
<tr>
<td>personalize</td>
<td>Can configure forms, lists, rules, controls, scripts.</td>
</tr>
<tr>
<td>personalize_choices</td>
<td>Can configure choices and predefined responses for non-journal fields designated as choice or suggestion fields.</td>
</tr>
<tr>
<td>personalize_control</td>
<td>Can configure controls on lists, such as filters, links, and buttons.</td>
</tr>
<tr>
<td>personalize_dictionary</td>
<td>Can configure dictionary entries and labels.</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>personalize_form</td>
<td>Can configure forms.</td>
</tr>
<tr>
<td>personalize_list</td>
<td>Can configure lists and list calculations.</td>
</tr>
<tr>
<td>personalize_responses</td>
<td>Can configure predefined responses for journal fields designated as suggestion fields.</td>
</tr>
<tr>
<td>personalize_rules</td>
<td>Can configure business rules and scripts. This role contains the following specialized roles for granting selective, administrative access to rules and scripts:</td>
</tr>
<tr>
<td></td>
<td>• business_rule_admin</td>
</tr>
<tr>
<td></td>
<td>• client_script_admin</td>
</tr>
<tr>
<td></td>
<td>• ui_policy_admin</td>
</tr>
<tr>
<td></td>
<td>• ui_action_admin</td>
</tr>
<tr>
<td>personalize_styles</td>
<td>Can configure field styles.</td>
</tr>
<tr>
<td>personalize_ui</td>
<td>Can configure forms and lists.</td>
</tr>
<tr>
<td>public</td>
<td>No login is required to access features or functions with the public role.</td>
</tr>
<tr>
<td>release_admin</td>
<td>Can edit Release history for a release.</td>
</tr>
<tr>
<td>report_admin</td>
<td>Can manage reports.</td>
</tr>
<tr>
<td>report_global</td>
<td>Can create global reports.</td>
</tr>
<tr>
<td>report_group</td>
<td>Can create reports and share reports with groups that the user is a member of. Users with this role can edit reports shared by other users in the group.</td>
</tr>
<tr>
<td>report_publisher</td>
<td>Can make reports available on a public page.</td>
</tr>
<tr>
<td>report_scheduler</td>
<td>Can schedule a report to be emailed.</td>
</tr>
<tr>
<td>script_fix_admin</td>
<td>Can manage fix scripts.</td>
</tr>
<tr>
<td>soap</td>
<td>Can query, create, update, and delete records on all tables, as well as execute scripts.</td>
</tr>
<tr>
<td>soap_create</td>
<td>Can create records on all tables and columns.</td>
</tr>
<tr>
<td>soap_delete</td>
<td>Can delete records on all tables and columns.</td>
</tr>
<tr>
<td>soap_ecc</td>
<td>Can query, create, and update on the ECC Queue table only.</td>
</tr>
<tr>
<td>soap_query</td>
<td>Can query records on all tables and columns.</td>
</tr>
<tr>
<td>soap_query_update</td>
<td>Can query and update records on all tables and columns.</td>
</tr>
<tr>
<td>soap_script</td>
<td>Can execute business rule endpoint function via script.do.</td>
</tr>
<tr>
<td>soap_update</td>
<td>Can update records on all tables and columns.</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>survey_admin</td>
<td>Can manage survey masters, questions, and instances. Contains the assessment admin role.</td>
</tr>
<tr>
<td>survey_reader</td>
<td>Can read survey instances and responses.</td>
</tr>
<tr>
<td>task_editor</td>
<td>Can edit protected task fields.</td>
</tr>
<tr>
<td>template_editor</td>
<td>Can edit or modify templates. Included in the itil role in the base system.</td>
</tr>
<tr>
<td>template_editor_global</td>
<td>Can create templates for global use.</td>
</tr>
<tr>
<td>template_editor_group</td>
<td>Can create templates for groups.</td>
</tr>
<tr>
<td>template_scheduler</td>
<td>Can schedule template-based record creation.</td>
</tr>
<tr>
<td>text_search_admin</td>
<td>Can customize Global Text Search groups and tables.</td>
</tr>
<tr>
<td>timecard_admin</td>
<td>Can approve, modify, and delete the time cards of other users.</td>
</tr>
<tr>
<td>ts_admin</td>
<td>Can administer Zing text search.</td>
</tr>
<tr>
<td>unlimited_createnow</td>
<td>Role for CreateNow unlimited licensed users.</td>
</tr>
<tr>
<td>user</td>
<td>Available for customer use, has no function in the base system.</td>
</tr>
<tr>
<td>user_admin</td>
<td>Can administer users, groups, locations, and companies.</td>
</tr>
<tr>
<td>view_changer</td>
<td>Can switch active views.</td>
</tr>
<tr>
<td>workflow_admin</td>
<td>Can create, edit, publish or delete graphical workflows.</td>
</tr>
<tr>
<td>workflow_creator</td>
<td>Can create new graphical workflows.</td>
</tr>
<tr>
<td>workflow_publisher</td>
<td>Can publish graphical workflows.</td>
</tr>
</tbody>
</table>

**Special administrative roles**

Certain roles grant specific administrative rights without the full privileges of the admin role. For example, an administrator can grant a user the right to change UI policy but not client scripts.

These roles do not change the behavior of the admin role, which grants full administrative privileges.

**Note:** The system does not support changing the name of any special administrative role. Changing the name of a special administrative role will prevent users and groups from accessing base system resources that depend on these roles.

<table>
<thead>
<tr>
<th>Role</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignment_rule_admin</td>
<td>Allows management of Assignment Rules.</td>
</tr>
<tr>
<td>ui_script_admin</td>
<td>Allows management of UI Scripts.</td>
</tr>
<tr>
<td>script_include_admin</td>
<td>Can manage Script Includes.</td>
</tr>
<tr>
<td>Role</td>
<td>Privilege</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>ui_page_admin</td>
<td>Can manage <strong>UI Pages</strong>.</td>
</tr>
<tr>
<td>ui_macro_admin</td>
<td>Can manage <strong>UI Macros</strong>.</td>
</tr>
<tr>
<td>form_admin</td>
<td>Can manage Forms, and Form Sections and Section Elements.</td>
</tr>
</tbody>
</table>

**Roles Contained in personalize_rules**

<table>
<thead>
<tr>
<th>Role</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>business_rule_admin</td>
<td>Can manage <strong>Business Rules</strong>.</td>
</tr>
<tr>
<td>client_script_admin</td>
<td>Can manage <strong>Client script types</strong> on page 3901.</td>
</tr>
<tr>
<td>ui_policy_admin</td>
<td>Can manage UI Policies.</td>
</tr>
<tr>
<td>ui_action_admin</td>
<td>Can Manage <strong>UI Actions</strong>.</td>
</tr>
</tbody>
</table>

**Read only role**

The read-only role (snc_read_only) restricts a user (or a group of users) to read-only access on the tables to which the user already has access.

This role is not intended to be the only role a user has. It is intended to be an additional role (in addition to the roles that the user normally has) for the purpose of restricting insert, update, and delete operations on the tables that the user can access as defined by the other roles.

After you assign this role to a user, they can no longer can create, update, or delete records on ANY tables.

**Note:** Assign this role only to users. Do not assign this role to other resources in the system, including applications, ACLs, and so on.

The snc_read_only role is included in the base ServiceNow system starting with the Fuji release and is active by default. Customers who upgrade to Fuji automatically have access to this role.

User administrators can assign the snc_read_only role to any user as a simple way to limit access to data without having to create ACLs for system and custom tables and fields. This is useful for performing internal or external audits without allowing a user to have insert or update access to data.

Users with the snc_read_only role have the following restrictions regardless of other roles and privileges they have.

- Cannot insert, update, or delete records from the UI or when using the GlideRecord API.
- Cannot activate or upgrade plugins.
- Cannot directly run SQL.
- Cannot upload XML files.
- Can only run background scripts when on an instance in the public sandbox environment.

**Note:** These role restrictions are in place even if impersonating another user with write access such as an admin.

**Activate the read only role**

An administrator can activate the Read Only User Role (com.snc.read_only.role) plugin if it is not already active.

Role required: admin

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
If the plugin depends on other plugins, these plugins are listed along with their activation status.

3. Optional: If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click **Activate**.

Read only role properties

   These system properties control the snc_read_only role. Administrators need to add these properties to the sys_properties table.

### Table 409: Read only role properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.security.snc_read_only_role.tables.exempt_create</td>
<td>Specifies which tables are exempt from the read-only role enforcement and allow the creation of new records.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: sys_user_session, sysevent, syslog, syslog_transaction, sys_user_preference, sys_ui_list, sys_ui_list_element, sys_db_cache</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.security.snc_read_only_role.tables.exempt_write</td>
<td>Specifies which tables are exempt from the read-only role enforcement and allow the updating of existing records.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: sys_user_session, sysevent, syslog, syslog_transaction, sys_user_preference, sys_ui_list, sys_ui_list_element, sys_db_cache</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.security.snc_read_only_role.tables.exempt_delete</td>
<td>Specifies which tables are exempt from the read-only role enforcement and allow the deletion of existing records.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: sys_user_preference, sys_ui_list, sys_ui_list_element, sys_db_cache</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>

Log on with the read only role
Users logging into a production instance should log in using the snc_read_only role to prevent unwanted modifications to the instance data.

1. Click To log on with different role(s), click here.
2. Enter read_only, maint.
3. Click Refresh.
4. Click read_only, maint to login.

Contextual security

The contextual security manager provides incredible flexibility and power to protect information by controlling read/write/create/delete authorization.

Key advantages

- **Contextual security**: Secure a record based on its contents
- **Hierarchical security**: Can apply security rules to any level in our object hierarchy

Everything you can do with the simple security manager you can also do with the contextual security manager. Likewise, after conversion to the contextual security manager, you should not see any behavior changes in your instance.

Securing Fields and Tables

Under the simple security manager, you could secure fields and tables by adding roles to the appropriate dictionary entry. After installing the contextual security manager, these dictionary roles are no longer tested. Instead the system looks for ACL rules on fields and or tables.

**Warning**: After you install the Contextual Security Manager you must secure fields and tables via ACL rules. Even if you configure the dictionary form and add roles to a dictionary entry, no change in rights will occur.

Granting Roles to Users

Roles can still be granted to users or groups using the same logic as under the simple security manager. The one noteworthy exception is that the "roles" field on the user record is no longer checked under the contextual security manager (and should be, in fact, removed from your user and group forms upon installation).

**Note**: To add roles to a user or group record under Contextual Security you must add them to the Roles related list instead of to the user or group record itself.

Applications and Modules

Applications and modules both contain lists of roles under which they can be viewed. For example, the System Definition application requires the admin role to be viewed.

Security rights for Applications and Modules are still defined via these role arrays although they may be transitioned to ACLs at some future date.
Catalog Items and Variables

Both catalog items and catalog variables contain lists of roles under which they can be viewed. Security rights for these entities are still defined via these role arrays although they may be transitioned to ACLs at some future date.

Inheritability of Group Roles

Under the contextual security manager, a group still automatically inherits any role granted to the group.

Note: The role's inherits flag is set to true.

Plugins

These plugins are automatically installed on new instances and can be activated for upgrades:

- **Contextual Security**: Provides contextual security functionality.
- **Contextual Security: Role Management Enhancements**: Eliminates duplicate inherited entries in the User Roles [sys_user_has_role] table (starting with the Geneva release) by tracking the number of times a role is inherited from another role or group. Once activated, inherited entries in the User Roles table are automatically managed and cannot be deleted directly. Instead, a containing role or group is expected to be removed from the user.

Rule search order

The system is aware of the instance object hierarchy when it tries to identify a security rule to apply to a particular entity in the contextual security model.

The search order for a field level rule is:

1. explicit rule on self
2. explicit rule on field in parent
3. ... until parent doesn't contain field
4. wildcard rule on self
5. wildcard rule on field in parent
6. ... until parent doesn't contain field

Example: Given incident.number

Search is:

1. incident.number
2. task.number
3. *.number
4. incident.*
5. task.*
6. **
Precedence between Row and Field Level Rules

What happens if a row level rule and a field level rule are in conflict? Perhaps my row level field indicates that I shouldn't be able to write to a particular row, but the field level rule indicates I do have write access?

In a nutshell, both rules must be met before an operation is allowed.

So, given a row level rule on incident, and a field level rule on incident.number, access to the number field would be allowed only if both rules evaluated to true.

Multiple Rules at the Same Level

What if the system, for example, finds two rules for incident.number?

The system will evaluate both rules and if either is true, then the requested access is allowed.

Category users and stakeholders

Users who take assessments are individuals in your organization who have specific knowledge of the assessment categories and the records being assessed. A person who is qualified to answer assessment questions from metrics in a specific category becomes a category user when associated to that category. A category user then becomes a stakeholder when associated to a specific assessable record.

The system determines which assessable records and questions a user receives by looking at the metric categories and assessable records for which those users are stakeholders. Assessment administrators can create and manage category users and stakeholders.

**Note:** Category users and stakeholders are used only for scheduled assessments.

Category users

Category users are users who can potentially answer assessment questions about a particular metric category for scheduled assessments. Category users should have special knowledge of the categories and the metrics those categories contain. A category user becomes a stakeholder when associated to an assessable record. Create at least one category user for each category you want to use for assessment questionnaires. There is no need for category users if the category contains only scripted metrics.

Stakeholders

A stakeholder is a category user with specific product or service knowledge, who is associated to an assessable record. A stakeholder is familiar with the assessable record and all the metrics within a specific category. The same category user can be associated to multiple assessable records, in which case the system creates a stakeholder record for each association.

There are multiple ways to create stakeholders:

- *Configure the system to create stakeholders automatically.*
- *Associate multiple category users to multiple assessable records.*
- *Associate multiple category users to one assessable record.*
- *Associate one category user to one assessable record.*

**Note:** Ensure you establish the desired user and assessable record associations to categories before attempting to creating stakeholders.
Designate a role delegator

Administrators (users with the admin role) can grant users the right to be role delegators. These delegators can assign roles to users who are in a particular group. The roles that delegators can assign to other users include the roles that the delegator inherits from a group those roles that the administrator specifies.

Users assigned the role_delegator role can act as delegators.

1. Navigate to **User Administration > Designate Role Delegator**.
2. Select the group that includes the user who you want to be the role delegator.
3. Select the user.

   ![Delegate roles](image)

4. Click **Submit**.

   A change request for the role delegator request is created and automatically approved.

   ![Change Request](image)

**Delegate roles**

To delegate specific roles to members of a group, navigate to **User Administration > Delegate Roles in Group** and fill out the form.

This module is available to users with the role_delegator role.
Figure 502: Role delegation

Table 410: Delegating Roles

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Select the group in which a member shall be delegated a role or roles. Any group can be selected, including groups that the role_delegator does not belong to or groups that the role_delegator does not manage.</td>
</tr>
<tr>
<td>User</td>
<td>Select the member who shall be delegated roles in that group.</td>
</tr>
<tr>
<td>Field</td>
<td>Input Value</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Roles to delegate</td>
<td>Select the roles to delegate to the group member. The roles available for delegating are only the roles that the role_delegator has.</td>
</tr>
</tbody>
</table>

Upon submission, a change request is created for the delegation request. This change request is approved automatically, and the specified roles are granted to the named user in the group selected.

Delegated roles can be removed in the same form by reversing the process. Select the group and user, remove the unwanted roles from the Roles slushbucket, and then re-submit the request.

*Designate a role delegator*

Administrators (users with the admin role) can grant users the right to be role delegates.

These delegates can assign roles to users who are in a particular group. The roles that delegates can assign to other users include the roles that the delegator inherits from a group those roles that the administrator specifies.

Users assigned the role_delegator role can act as delegates.

1. Navigate to **User Administration > Designate Role Delegator**.
2. Select the group that includes the user who you want to be the role delegator.
3. Select the user.

4. Click **Submit**.

A change request for the role delegator request is created and automatically approved.

*View delegated roles*

An administrator can view role designation in the following locations.

- User records
- Role Delegates module
- Role Audit module
User Records

Open a user record by navigating to User Administration > Users and selecting the user. You can see all the roles assigned to that user in the Roles related list.

Figure 503: List of role delegators

Role Delegates

To view existing role delegators and the groups in which they can delegate roles, navigate to User Administration > Role Delegates. All the role delegators and the groups they belong to are listed.

Figure 504: Role Delegator List

Role Audit

The Audit Role list view displays all the role changes made in the instance by user and group. To access the Audit Role list, navigate to System Security > Reports > Role Audit.

Prevent a role from being delegated
You can prevent roles from being delegated to users.

By default, the following roles cannot be delegated.
• admin
• public
• nobody
• role_delegator, a user with the role_delegator role cannot delegate this role to other group members

1. Navigate to User Administration > Roles.
2. Open the role.
3. Configure the form to add the Grantable or Can delegate fields.
4. Clear the check box for one or both of these fields.
5. Click Update.

Role delegation and record producers
The Role Delegation modules link to record producers. These record producers create change requests that are automatically approved by graphical workflows.

These graphical workflows include the following:
• Grant role_delegator role to user in group
• Delegate roles to group member

These workflows can be customized as desired to add approval steps.

Group manager change business rule
The Group Manager Change business rule, which is disabled by default, automatically grants the role_delegator role to a user who is designated manager of a group in the Manager field on the Group form. The role is removed when the user is no longer the manager of the group.

Activate the business rule to take advantage of it.

Security jump start - ACL rules
The Security Jump Start (ACL Rules) Plugin is installed automatically on all new instances.

Note:
• Plugin Required
• Functionality described here requires the Security Jump Start (ACL Rules) plugin. The plugin is automatically installed for new instances.

These rules were written to provide a jump start on securing many system tables, to make it easier for an organization to more quickly get into production.

This plugin is not intended for existing instances, as it might modify security access to tables that are already in use in a production environment. If an admin is interested in the new ACL rules provided by this plugin, one or more of them may be created manually in an existing instance as specific needs dictate. This list of ACLs may be used as a guideline in that case. Should an admin strongly want this plugin installed on an existing instance, we highly recommend the plugin be tested extensively in a test instance first, to ensure that the rules do not conflict with the operational needs of the organization's current implementation.

The following ACLs are included in this plugin. Click the icon in a header row to sort that column in ascending or descending order. The Operation key is as follows:
• R=read
• W=write
- D=delete
- C=create

<table>
<thead>
<tr>
<th>Name</th>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmdb_ci</td>
<td>WCD</td>
<td>asset or itil role required to write/create/delete Configuration Item records</td>
</tr>
<tr>
<td>cmn_department</td>
<td>WD</td>
<td>user_admin role required to write/delete Department records</td>
</tr>
<tr>
<td>cmn_location</td>
<td>WC</td>
<td>user_admin role required to write/create Location records</td>
</tr>
<tr>
<td>core_company</td>
<td>WD</td>
<td>user_admin role required to write/delete Company records</td>
</tr>
<tr>
<td>kb_knowledge</td>
<td>create</td>
<td>knowledge role required to created Knowledge records</td>
</tr>
<tr>
<td>ldap_ou_config</td>
<td>RWCD</td>
<td>user_admin role required to read/write/create/delete LDAP OU Definition records</td>
</tr>
<tr>
<td>ldap_server_config</td>
<td>RWCD</td>
<td>user_admin role required to read/write/create/delete LDAP Server records</td>
</tr>
<tr>
<td>process_guide</td>
<td>WCD</td>
<td>admin role required to write/create/delete Process Guide records</td>
</tr>
<tr>
<td>process_step</td>
<td>WCD</td>
<td>admin role required to write/create/delete Process Step records</td>
</tr>
<tr>
<td>sc_category</td>
<td>create</td>
<td>catalog_admin role required to create Service Catalog Category records</td>
</tr>
<tr>
<td>sc_category</td>
<td>delete</td>
<td>catalog_admin role required to delete Service Catalog Category records</td>
</tr>
<tr>
<td>sc_category</td>
<td>write</td>
<td>catalog_admin role required to write to Service Catalog Category records</td>
</tr>
<tr>
<td>sc_cat_item</td>
<td>write</td>
<td>catalog_admin role required to write to Catalog Item records</td>
</tr>
<tr>
<td>sc_cat_item</td>
<td>delete</td>
<td>catalog_admin role required to delete Catalog Item records</td>
</tr>
<tr>
<td>sc_cat_item</td>
<td>create</td>
<td>catalog_admin role required to create Catalog Item records</td>
</tr>
<tr>
<td>sysevent_email_action</td>
<td>read</td>
<td>all users can read Email Notification records (for subscription purposes)</td>
</tr>
<tr>
<td>Name</td>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>sysevent_register</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete Event Registry records</td>
</tr>
<tr>
<td>sysevent_script_action</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete Script Action records</td>
</tr>
<tr>
<td>syslog</td>
<td>RWCD</td>
<td>admin required to read/write/create/delete Log Entry records</td>
</tr>
<tr>
<td>sysrule</td>
<td>RWCD</td>
<td>admin required to read/write/create/delete Rule records (Email Notifications, Inbound Email Actions, Approval Rules, etc.)</td>
</tr>
<tr>
<td>sysrule</td>
<td>read</td>
<td>all users can read Email Notification records for (subscription based notifications)</td>
</tr>
<tr>
<td>sys_app_application</td>
<td>WCD</td>
<td>admin required to write/create/delete Application records</td>
</tr>
<tr>
<td>sys_app_category</td>
<td>WCD</td>
<td>admin role required to write/create/delete Application Category records</td>
</tr>
<tr>
<td>sys_app_module</td>
<td>WCD</td>
<td>admin required to write/create/delete Module records</td>
</tr>
<tr>
<td>sys_audit</td>
<td>RWCD</td>
<td>admin required to read/write/create/delete Audit records</td>
</tr>
<tr>
<td>sys_dictionary</td>
<td>RWC</td>
<td>personalize_dictionary role required to read/write/create Dictionary records</td>
</tr>
<tr>
<td>sys_dictionary.*</td>
<td>read</td>
<td>personalize_dictionary role can read Dictionary fields</td>
</tr>
<tr>
<td>sys_documentation</td>
<td>delete</td>
<td>personalize_dictionary role required to delete Field Label records</td>
</tr>
<tr>
<td>sys_documentation</td>
<td>create</td>
<td>personalize_dictionary role required to create Field Label records</td>
</tr>
<tr>
<td>sys_documentation</td>
<td>write</td>
<td>personalize_dictionary role required to write to Field Label records</td>
</tr>
<tr>
<td>sys_gauge</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete Gauge records</td>
</tr>
<tr>
<td>Name</td>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sys_gauge_count</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete Gauge Count records</td>
</tr>
<tr>
<td>sys_group_has_role</td>
<td>read</td>
<td>itil role required to see Group Role records</td>
</tr>
<tr>
<td>sys_home</td>
<td>WCD</td>
<td>itil admin role required to write/create/delete Welcome Page Section records</td>
</tr>
<tr>
<td>sys_installation_exit</td>
<td>WCD</td>
<td>admin role required to write/create/delete Installation Exit records</td>
</tr>
<tr>
<td>sys_job</td>
<td>WCD</td>
<td>admin role required to write/create/delete Sys Job records</td>
</tr>
<tr>
<td>sys_nav_link</td>
<td>WCD</td>
<td>admin role required to write/create/delete Navigation Link records</td>
</tr>
<tr>
<td>sys_perspective</td>
<td>WCD</td>
<td>admin role required to write/create/delete Menu List records</td>
</tr>
<tr>
<td>sys_portal</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete Portal records</td>
</tr>
<tr>
<td>sys_portal_page</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete Homepage records</td>
</tr>
<tr>
<td>sys_portal_preferences</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete Portal Preferences records</td>
</tr>
<tr>
<td>sys_processor</td>
<td>WC</td>
<td>admin role required to write/create Processor records</td>
</tr>
<tr>
<td>sys_properties</td>
<td>WC</td>
<td>admin role required to write/create System Property records</td>
</tr>
<tr>
<td>sys_properties_category</td>
<td>WCD</td>
<td>admin role required to write/create/delete Property Category records</td>
</tr>
<tr>
<td>sys_report</td>
<td>delete</td>
<td>roles that can delete Report records (does not restrict deleting through Report UI)</td>
</tr>
<tr>
<td>sys_report</td>
<td>write</td>
<td>roles that can write to Report records (does not restrict editing through Report UI)</td>
</tr>
<tr>
<td>sys_report</td>
<td>read</td>
<td>users can read their own Report records, those of their groups, and GLOBAL ones (does not affect viewing through Report UI)</td>
</tr>
<tr>
<td>Name</td>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sys_report</td>
<td>read</td>
<td>roles that can read Report records (does not restrict viewing through Report UI)</td>
</tr>
<tr>
<td>sys_reportroles</td>
<td>read</td>
<td>admin role required to read Report Roles records</td>
</tr>
<tr>
<td>sys_script</td>
<td>WCD</td>
<td>admin role required to write/create/delete Business Rule records</td>
</tr>
<tr>
<td>sys_script_ajax</td>
<td>WCD</td>
<td>admin role required to write/create/delete AJAX Script records</td>
</tr>
<tr>
<td>sys_script_client</td>
<td>WCD</td>
<td>admin role required to write/create/delete Client Script records</td>
</tr>
<tr>
<td>sys_script_include</td>
<td>WCD</td>
<td>admin role required to write/create/delete Script Include records</td>
</tr>
<tr>
<td>sys_security_acl</td>
<td>write</td>
<td>admin role required to write to Access Control records</td>
</tr>
<tr>
<td>sys_security_acl_role</td>
<td>create</td>
<td>admin role required to create Access Roles records</td>
</tr>
<tr>
<td></td>
<td>delete</td>
<td>admin role required to delete Access Roles records</td>
</tr>
<tr>
<td></td>
<td>write</td>
<td>admin role required to write to Access Roles records</td>
</tr>
<tr>
<td>sys_security_operation</td>
<td>delete</td>
<td>admin role required to delete Security Operation records</td>
</tr>
<tr>
<td></td>
<td>create</td>
<td>admin role required to create Security Operation records</td>
</tr>
<tr>
<td></td>
<td>write</td>
<td>admin role required to write to Security Operation records</td>
</tr>
<tr>
<td>sys_security_type</td>
<td>write</td>
<td>admin role required to write to Security Type records</td>
</tr>
<tr>
<td></td>
<td>create</td>
<td>admin role required to create Security Type records</td>
</tr>
<tr>
<td></td>
<td>delete</td>
<td>admin role required to delete Security Type records</td>
</tr>
<tr>
<td>sys_status</td>
<td>create</td>
<td>admin role required to create System Status records</td>
</tr>
<tr>
<td></td>
<td>delete</td>
<td>admin role required to delete System Status records</td>
</tr>
<tr>
<td></td>
<td>write</td>
<td>admin role required to write to System Status records</td>
</tr>
<tr>
<td>Name</td>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>sys_template</td>
<td>write</td>
<td>template_editor role required to write to Template records</td>
</tr>
<tr>
<td>sys_template</td>
<td>create</td>
<td>template_editor role required to create Template records</td>
</tr>
<tr>
<td>sys_template</td>
<td>delete</td>
<td>template_editor role required to delete Template records</td>
</tr>
<tr>
<td>sys_template</td>
<td>read</td>
<td>template_editor role required to read Template Roles records</td>
</tr>
<tr>
<td>sys_ui_action</td>
<td>create</td>
<td>admin role required to create UI Action records</td>
</tr>
<tr>
<td>sys_ui_action</td>
<td>delete</td>
<td>admin role required to delete UI Action records</td>
</tr>
<tr>
<td>sys_ui_action</td>
<td>write</td>
<td>admin role required to write to UI Action records</td>
</tr>
<tr>
<td>sys_ui_action_view</td>
<td>write</td>
<td>admin role required to write to UI View Action records</td>
</tr>
<tr>
<td>sys_ui_action_view</td>
<td>create</td>
<td>admin role required to create UI View Action records</td>
</tr>
<tr>
<td>sys_ui_action_view</td>
<td>delete</td>
<td>admin role required to delete UI View Action records</td>
</tr>
<tr>
<td>sys_ui_policy</td>
<td>create</td>
<td>admin role required to create UI Policy records</td>
</tr>
<tr>
<td>sys_ui_policy</td>
<td>delete</td>
<td>admin role required to delete UI Policy records</td>
</tr>
<tr>
<td>sys_ui_policy</td>
<td>write</td>
<td>admin role required to write to UI Policy records</td>
</tr>
<tr>
<td>sys_ui_policy_action</td>
<td>create</td>
<td>admin role required to create UI Policy Action records</td>
</tr>
<tr>
<td>sys_ui_policy_action</td>
<td>delete</td>
<td>admin role required to delete UI Policy Action records</td>
</tr>
<tr>
<td>sys_ui_policy_action</td>
<td>write</td>
<td>admin role required to write to UI Policy Action records</td>
</tr>
<tr>
<td>sys_ui_script</td>
<td>write</td>
<td>admin role required to write to UI Script records</td>
</tr>
<tr>
<td>sys_ui_script</td>
<td>delete</td>
<td>admin role required to delete UI Script records</td>
</tr>
<tr>
<td>sys_ui_script</td>
<td>create</td>
<td>admin role required to create UI Script records</td>
</tr>
<tr>
<td>sys_user</td>
<td>write</td>
<td>Users with no role cannot update any user record but their own</td>
</tr>
<tr>
<td>Name</td>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sys_user_grmember</td>
<td>delete</td>
<td>user_admin role required to delete Group Member records</td>
</tr>
<tr>
<td>sys_user_grmember</td>
<td>write</td>
<td>user_admin role required to write to Group Member records</td>
</tr>
<tr>
<td>sys_user_group</td>
<td>create</td>
<td>Only itil and above can create group records</td>
</tr>
<tr>
<td>sys_user_group</td>
<td>write</td>
<td>Only itil and above can write to group records</td>
</tr>
<tr>
<td>sys_user_has_role</td>
<td>read</td>
<td>itil role required to see User Role records</td>
</tr>
<tr>
<td>sys_user_role</td>
<td>create</td>
<td>admin role required to create Role records</td>
</tr>
<tr>
<td>sys_user_role</td>
<td>delete</td>
<td>admin role required to delete Role records</td>
</tr>
<tr>
<td>sys_user_role</td>
<td>write</td>
<td>admin role required to write to Role records</td>
</tr>
<tr>
<td>sys_user_role_contains</td>
<td>read</td>
<td>itil role required to see Contained Role records</td>
</tr>
<tr>
<td>sys_user_role_contains</td>
<td>write</td>
<td>admin role required to write to Contained Role records</td>
</tr>
<tr>
<td>sys_user_token</td>
<td>RWCD</td>
<td>admin role required to read/write/create/delete User Token records</td>
</tr>
</tbody>
</table>

Audit user roles

Changes to user roles are automatically tracked in the Audit Roles [sys_audit_role] table.

Role required: admin

**Note:** If the Contextual Security: Role Management Enhancements on page 2508 plugin is installed, you must Enable role auditing with Contextual Security: Role Management Enhancements on page 2510.

Navigate to the Audit Roles [sys_audit_role] table.

The Audit Roles table displays changes to user roles.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed by</td>
<td>The user that made the change.</td>
</tr>
<tr>
<td>Count after change</td>
<td>Direct role added as a result of the change (if any) plus the number of inherited roles added.</td>
</tr>
<tr>
<td>Granted by group</td>
<td>If the role was inherited, the group that the role was inherited from.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Operation</td>
<td>The type of change. Values include:</td>
</tr>
<tr>
<td></td>
<td>• Added</td>
</tr>
<tr>
<td></td>
<td>• Removed</td>
</tr>
<tr>
<td>Role</td>
<td>The affected role.</td>
</tr>
<tr>
<td>User</td>
<td>The affected user.</td>
</tr>
</tbody>
</table>

**Groups**

A group is a set of users who share a common purpose.

Groups may perform tasks such as approving change requests, resolving incidents, receiving email notifications, or performing work order tasks. Any business rules, assignment rules, system roles, or attributes that refer to the group apply to all group members automatically. When you add users, make sure that each one is associated with a group. Users with the user_admin role can create and edit groups.

**Create a user**

You can add a user to ServiceNow by navigating to **User Administration > Users**.

1. Navigate to **User Administration > Users**.
2. Click **New**.
3. Enter the user's information (see *table*).

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td>Create a unique identifier for this user's ServiceNow login user name. Typical examples of user IDs are cwitherspoon and charlie.witherspoon. You cannot create a new user whose User ID duplicates an existing user. If you do import duplicates from an update set, the more recently created names takes the duplicate User ID.</td>
</tr>
<tr>
<td>First name</td>
<td>Enter the user's full first name.</td>
</tr>
<tr>
<td>Last name</td>
<td>Enter the user's last name.</td>
</tr>
<tr>
<td>Title</td>
<td>Enter a title or job description, or select one from the list.</td>
</tr>
<tr>
<td>Department</td>
<td>Select the user's department from the list.</td>
</tr>
<tr>
<td>Password</td>
<td>Assign a password to the user. This password can be permanent or temporary.</td>
</tr>
<tr>
<td>Password needs reset</td>
<td>Select this check box to require the user to change the password during the first login.</td>
</tr>
<tr>
<td>Locked out</td>
<td>Select this check box to lock the user out of the instance and terminate all their active sessions. The system prevents users with the admin role from locking themselves out.</td>
</tr>
<tr>
<td>Field</td>
<td>Input Value</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make this user active. Only the administrator sees inactive user in:</td>
</tr>
<tr>
<td></td>
<td>• Lists of users</td>
</tr>
<tr>
<td></td>
<td>• The selection list on reference fields (magnifying glass icon)</td>
</tr>
<tr>
<td></td>
<td>• The auto-complete list that appears when you type into a reference field</td>
</tr>
<tr>
<td>Web service access only</td>
<td>Select this check box to designate this user as a non-interactive user. This field is available with Non-Interactive Sessions.</td>
</tr>
<tr>
<td>Internal Integration User</td>
<td>Select this check box to designate this user as an internal integration user.</td>
</tr>
<tr>
<td>Date format</td>
<td>Select the user's preferred format for dates.</td>
</tr>
<tr>
<td>Email</td>
<td>Enter the user's email address.</td>
</tr>
<tr>
<td></td>
<td>To enter a non-standard email address that does not pass field validation, you must deactivate the validation script first.</td>
</tr>
<tr>
<td></td>
<td>1. Navigate to System Definition &gt; Validation Scripts.</td>
</tr>
<tr>
<td></td>
<td>2. Select the email record.</td>
</tr>
<tr>
<td></td>
<td>3. Clear the Active check box and save the change.</td>
</tr>
<tr>
<td></td>
<td>4. Complete the user profile, including the email address, and update or submit the record.</td>
</tr>
<tr>
<td></td>
<td>5. Reactivate the email validation script.</td>
</tr>
<tr>
<td>Notification</td>
<td>Select the type of notification to send to this user. The default is Email. If you select None, the user can still receive notifications if he or she subscribes to the notification or is specified as a recipient in the Email Notifications form.</td>
</tr>
<tr>
<td></td>
<td>To prevent notification completely, set a condition on the Email Notification form itself that does not deliver the notification if this field is set to None.</td>
</tr>
<tr>
<td>Calendar integration</td>
<td>Select Outlook to have this user receive meeting notifications via email directly to the calendar. Otherwise, select None.</td>
</tr>
<tr>
<td>Time zone</td>
<td>Select the user's time zone.</td>
</tr>
<tr>
<td>Business phone</td>
<td>Enter this user's business phone number.</td>
</tr>
<tr>
<td>Field</td>
<td>Input Value</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Mobile phone</td>
<td>Enter this user's mobile phone number.</td>
</tr>
<tr>
<td>Photo</td>
<td>Attach a photo of the user, if appropriate.</td>
</tr>
<tr>
<td>Geolocation tracked</td>
<td>Select the check box to enable location tracking. The Geolocation tracked field, which is available when Geolocation is activated, provides the option to track a user's location.</td>
</tr>
<tr>
<td>Location</td>
<td>Select the user's usual location. This field is visible when geolocation is active.</td>
</tr>
</tbody>
</table>

4. Optional: *Customize the form* to add the **Schedule** field and assign a *schedule* to the user.

5. **Click Submit.**

   The new user record appears at the top of the list.

**Allow a user to view a profile**

Users are able to view their profile by clicking their name in the Welcome banner. If your users cannot do this, enable a system property.

1. **Navigate to the System Properties [sys_properties] table.**
2. **Search for the glide.ui.welcome.profile_link property.**
3. **Set the value to true.**

**Deactivate the validation script during user creation**

To enter a non-standard email address that does not pass field validation, you must deactivate the validation script first.

1. **Navigate to System Definition > Validation Scripts.**
2. **Select the email record.**
3. **Clear the Active check box and save the change.**
4. **Complete the user profile, including the email address, and update or submit the record.**
5. **Reactivate the email validation script.**

**Add a new company**

How to add a new company.

1. **Navigate to User Administration > Companies.**
2. **Click New.**
3. **Complete the fields, as appropriate (See table).**
Table 411: Adding A Company

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the company.</td>
</tr>
<tr>
<td>Phone</td>
<td>Company phone number.</td>
</tr>
<tr>
<td>Fax phone</td>
<td>Company fax number.</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Whether the company is a manufacturer.</td>
</tr>
<tr>
<td>Vendor</td>
<td>Whether the company is a vendor.</td>
</tr>
<tr>
<td>Stock symbol</td>
<td>Three or four letter stock symbol for the company.</td>
</tr>
<tr>
<td>Stock price</td>
<td>Current price at which company stock is sold.</td>
</tr>
<tr>
<td>Street</td>
<td>Mailing street address of the company.</td>
</tr>
<tr>
<td>City</td>
<td>City in which the company is located.</td>
</tr>
<tr>
<td>State</td>
<td>State or province in which the company is located.</td>
</tr>
<tr>
<td>Zip/Postal code</td>
<td>Zip or postal code for the company.</td>
</tr>
<tr>
<td>Notes</td>
<td>Any information about the company that would be helpful for others to know.</td>
</tr>
</tbody>
</table>

Fields that can be added by configuring the form:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude</td>
<td>The latitude of the company, if applicable. This field is populated by a business rule called get_lat_long. Deactivate this business rule to prevent the system from overwriting any values populated in the field manually. Latitude is expressed as a floating point data type. Latitude from upgraded versions of ServiceNow expressed in any format other than floating point appears in a column called Old Latitude. The system attempts to convert all latitude values from previous versions to the floating point notation, where possible.</td>
</tr>
<tr>
<td>Longitude</td>
<td>The longitude of the company, if applicable. This field is populated by a business rule called get_lat_long. Deactivate this business rule to prevent the system from overwriting any values populated in the field manually. Longitude is expressed as a floating point data type. Longitude from upgraded versions of ServiceNow expressed in any format other than floating point appears in a column called Old Longitude. The system attempts to convert all longitude values from previous versions to the floating point notation, where possible.</td>
</tr>
</tbody>
</table>

**Note:** The IT Finance application adds a Finance view to the Company form. The Finance view adds a chart that shows expenses that were allocated to the company.

---

**Add a new department**

Departments are another way to categorize users, groups, and assets.

1. From the left navigation pane, select **User Administration > Departments**.
2. Click **New** to add a new department.
3. When done, click **Submit**.
Associate a user to a group

You can associate users with a group.

1. Navigate to User Administration > Groups.
2. Click the group to which you want to assign the user.
3. In the Group Members related list, click Edit.
4. Select the user in the Collection list, and then click Add.
5. Click Save.

Non-interactive sessions

The Non-Interactive Sessions plugin creates a distinction between interactive and non-interactive users.

- Interactive users can log in to the ServiceNow UI and can use their credentials for SOAP connections if allowed by strict security. They can use their credentials for other API connections such as WSDL, JSON, XML, or XSD without restriction.
- Non-interactive users can only use their credentials to authorize API connections such as JSON, SOAP, and WSDL. They cannot log in to the ServiceNow UI. The strict security high security setting determines if non-interactive users are subject to Contextual Security requirements.

Distinguishing between interactive and non-interactive users increases instance security by ensuring that users conform to the principle of least privilege.

Installed with Non-Interactive Sessions

The Non-Interactive Sessions plugin installs certain changes.

- Adds a column Web Service Access Only [web_service_access_only] to the User [sys_user] table.
- Changes all existing users to be interactive users (web_service_access_only=false).
- Updates the User form to display the Web Service Access Only [web_service_access_only] field by default.

Create an interactive user

Interactive users have the following access rights.

They can:

- Use their user name and password to log in to the UI or a content management system (CMS) portal.
• Connect to an instance from a URL that calls a **UI page**, form, or list (for example, https://<instance name>.service-now.com/incident.do).
• Connect with single sign-on (for example, digest authentication or SAML).
• Use their credentials to authorize SOAP connections if allowed by **strict security**.
• Use their credentials to authorize any other type of API connection without restriction.

When you activate the Non-Interactive Sessions plugin all existing users automatically become interactive users. New users default to interactive users unless you manually make them non-interactive. Use the following steps to manually switch a non-interactive user back to an interactive user.

1. Navigate to **User Administration > Users**.
2. Search for the user you want to update. For example, System Administrator.
3. Clear the **Web Service Access Only** check box.
4. Click **Update**.

**Create a non-interactive user**
Non-interactive users can only connect to ServiceNow from an API protocol.
They cannot:
• Use their user name and password to **log in** to the UI or a content management system (CMS) portal.
• Connect to an instance from a URL that calls a **UI page**, form, or list (for example, https://<instance name>.service-now.com/incident.do).
• Connect with single sign-on (for example, digest authentication or SAML).
• Be used as the MID Server user.

After installing the Non-Interactive Sessions plugin, consider updating your existing web service user accounts to be non-interactive users.

1. Navigate to **User Administration > Users**.
2. Search for the user to be updated. For example, SOAP user.
3. Select the **Web Service Access Only** check box.
4. Click **Update**.

**Note:**
ServiceNow always uses any user name and password credentials supplied with a request even if the **High Security Settings** do not require authorization for a given API protocol. For example, if a SOAP request supplies a user name and password, the instance verifies those credentials even if SOAP requests do not require authorization. To avoid verifying user credentials, the request must not include them.

**Update web service user accounts for strict security**
If your instance requires strict security, add the soap role to any user accounts used for web services.

1. Navigate to **User Administration > Users**.
2. Select a web service user from the list.
3. From the **Roles** related list, click **Edit**.
4. Add soap to the **Roles List**.
5. Click **Save**.
6. Click **Update**.
Request the Non Interactive Sessions plugin

The Non-Interactive Sessions feature is enabled for all new instances starting with the Calgary release. For instances that upgraded from a previous release, you can request the Non Interactive Sessions plugin (com.glide.security.non_interactive_users) through the HI Customer Service system.

Role required: none

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
<tr>
<td>Specify the date and time you would like this plugin to be enabled</td>
<td>Date and time must be at least 2 business days from the current time.</td>
</tr>
<tr>
<td>Reason/Comments</td>
<td>Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

Require authentication

You can specify whether non-interactive connections require authentication from the **High Security Settings** module.

A non-interactive connection bypasses the UI to connect to the instance at an API level. Typically, non-interactive connections use set protocols such as JSON, SOAP, XSD, or WSDL. By default, all non-interactive connections require authentication.

1. Login with an administrator user with the security_admin role.
2. Elevate your privileges to use security_admin.
3. Navigate to **System Security > High Security Settings**.
4. Select the matching "Requires authorization" option for the protocol you want to set. For example, **Requires authorization for incoming SOAP requests**.
5. Select the checkbox to require authentication for the non-interactive connection method. Clear the checkbox to allow the non-interactive connection method to connect without providing any credentials.

---

**Note:** Activating the Non-Interactive Sessions plugin on an existing system may prevent any existing users that authorize SOAP and WSDL-based integrations from logging in unless they already have the soap role. See [Updating Web Service User Accounts for Strict Security](#) to manually update existing integration users.
Create a group

Users with the user_admin role can create groups.

1. Navigate to User Administration > Groups.
2. Click New.
3. Fill in the form.

To see some of the fields, you may need to personalize the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the group.</td>
</tr>
<tr>
<td>Manager</td>
<td>Group manager or lead.</td>
</tr>
<tr>
<td>Type</td>
<td>Category for this group. For example, a group designated as type catalog is a service catalog group and can also be accessed under the Service Catalog &gt; Catalog Policy &gt; Fulfillment Groups module. You may need to personalize the form to add the Type field. Activating the Work Management plugin adds the Type field automatically.</td>
</tr>
<tr>
<td>Group email</td>
<td>Group email distribution list or the email address of the group's point of contact, such as the group manager.</td>
</tr>
<tr>
<td>Parent</td>
<td>Other group of which this group is a member. If a group has a parent, the child group inherits the roles of the parent group. The members of the child group are not members of the parent group. For example, if an incident is assigned to the parent group and you click the Assigned to lookup icon, only the members in the parent group are available. The members of the child group are not available.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that indicates whether the group is active or inactive. Inactive groups still appear in any reference field that already references the group, but are not visible by non-admin users in: • lists of groups • the reference lookup list for reference fields • the auto-complete list of groups displayed when you type into a reference field</td>
</tr>
<tr>
<td>Exclude manager</td>
<td>Check box that controls whether the group's manager receives email notifications.</td>
</tr>
<tr>
<td>Include members</td>
<td>Check box that controls whether the group members receive individual emails when someone sends an email to the Group Email address. The only exception to this functionality is for approval notifications, whereby all members</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Description</td>
<td>Helpful information about the group.</td>
</tr>
</tbody>
</table>

**Add a user to a group**

Users with the user_admin role can add a user to a group.

1. Navigate to User Administration > Groups.
2. Click a group Name.
3. In the Group Members related list, click Edit.
4. Select one or more names in the Collection list.
5. Click Add.
6. Click Submit.

**Remove a user from a group**

Users with the user_admin role can remove a user from a group.

1. Navigate to User Administration > Groups.
2. Click a group Name.
3. In the Group Members related list, select the check box next to a group member name.
4. From the Actions on selected rows menu, select Delete.

**Hide groups**

You can hide groups by introducing a hidden field to the group form and then creating a business rule to filter out groups marked as "hidden."

Only users in the hidden group will be able to see the hidden field when selecting a group in a reference field.

**Add a business rule to filter out hidden groups**

This business rule filters out hidden groups. Only users within the hidden group and users marked admin or groups_admin can see the group.

Create a new "before query" business rule on the sys_user_group (Group) table with the following script:

```javascript
if (!gs.hasRole("admin") && !gs.hasRole("groups_admin") &&
    gs.getSession().isInteractive()) {
    var qc = current.addQuery("u_hidden", "!=", "true"); //cannot see hidden groups...
    qc.addOrCondition("sys_id", "javascript:getMyGroups()" ); //...unless in the hidden group
}
```

**Add a "hidden" field to a group form**

To add a hidden field to a group form:

Create a new true/false field labeled Hidden on the Group form.

The system creates new field called u_hidden on the sys_user_group table and enables use of the Hidden check box to designate a hidden group.

**Assign a role to a user**

You can assign roles to a user by navigating to User Administration > Users.
A user automatically inherits roles from all groups the user belongs to. These roles cannot be deleted from the user's record, only from the group's record. Roles can also be associated directly with the user.

To add roles to a user's record:

1. Navigate to **User Administration > Users**.
2. Open a user's record.
3. In the **Roles** related list, click **Edit**.
4. Select the desired roles in the **Collection** list, and then click **Add**.
5. Click **Save**.

**Configure group types for assignment groups**

The **Type** field can be used to define categories of groups, which can be used to filter out assignment groups based on the group type using a reference qualifier on [task.assignment_group].

For example, when selecting an assignment group from the Incident form, **Type** can be used to filter groups based on whether they are typically involved in the Incident Management process. Groups such as Network or Help Desk are displayed as they are typically involved. Groups such as HR or New York are omitted.

The following items are provided in the base system.

- The types **catalog**, **itil**, and **survey**.
- The reference qualifier on [task.assignment_group] filters on **[Type] equals [none]**.
- A reference qualifier named GetGroupFilter is available to filter for group types.

**Note:** Dictionary overrides allow administrators to filter for a group type on an extended table with a simple reference qualifier override.

**Add a new group type**

You can add additional group types to filter assignment groups for tasks.

Role required: user_admin or admin

You may need to configure the form to display the **Type** field.

1. Navigate to **User Administration > Groups**.
2. Select a group record.
3. Click the lock icon beside **Type**.
4. Click the lookup icon beside the selection field.
   The Group Types dialog opens.
5. Complete the following steps.
   a) Click **New**.
   b) Enter the group type name and description.
      For example, to define a group's type as incident and problem, enter: **incident,problem**.
      Click **Submit**.

      The Group form reopens with the new type listed.
6. Optional: Add additional group types if needed.
7. Click **Update**.

**Assign a group type**

You can assign group types to filter assignment groups for tasks.
Role required: user_admin or admin

1. Navigate to User Administration > Groups and select the desired group.
2. Click the lock icon beside Type.
3. Click the lookup icon beside the selection field and select one or more group types.

   Note: Because the default behavior of [task.assignment_group] is to filter out groups with group types defined, adding a type to a group filters it out of the Assignment Group field on tasks. To change the behavior, set up the reference qualifier.

4. Click Update.

Reference qualifier setup
The reference qualifier on [task.assignment_group] filters which groups are available to be used as an assignment group.

Changing the reference qualifier on [task.assignment_group] affects every table which extends the Task Table. Dictionary overrides can be used to define reference qualifiers on tables that extend Task without affecting other task tables.

There are two ways to use reference qualifiers to filter assignment group types, simple or advanced reference qualifiers.

Simple reference qualifier
In the base system, the simple reference qualifier type=null is used to allow groups with no defined type to be selected. In the same way, if there is one specific type which should be available from the Assignment Group, specify it in the same format. For example, if there is a group type assignment which all assignment groups have, use the reference qualifier type=assignment to return only those groups.

Advanced reference qualifiers
If a simple reference qualifier doesn't provide enough control over filtering, an advanced filtering function can be defined in a business rule and called using advanced reference qualifiers. The business rule GetGroupFilter (available in the base system) can be called by an advanced reference qualifier to filter group types based on arguments.

For example, the following reference qualifier restricts the choice list to groups with type database or network:

```
javascript:GetGroupFilter('database,network')
```

The GetGroupFilter function can also be called from reference qualifier functions which you create.

⚠️ Warning: Exercise caution if the tree picker is used in conjunction with filtering by group type. The reference qualifier function must select groups at all levels of the tree because a group is only selectable if its parent is selectable.

Create an advanced reference qualifier
For more advanced filtering, create an Advanced Reference Qualifier for the Assignment Group field on the Task table.

For more advanced filtering, create an advanced reference qualifier for the Assignment Group field on the Task [task] table. Note that this reference qualifier will apply to all tables based on [task], unless dictionary overrides are used.
The following example reference qualifier assumes that you have created a group type named database. If the task is an incident record and the category is Database, then the reference qualifier will only select database groups. In all other circumstances it will select any group which is not a catalog group.

```javascript
function TaskAssignmentFilter() {
  var classname = current.getRecordClassName();
  var filter = "type=null";
  if (classname == "incident" && current.category == "database") {
    filter = GetGroupFilter("database");
  } else {
    // append 'catalog' group exclusion to the filter
    var cat = new GlideRecord("sys_user_group_type");
    cat.addQuery("name", "catalog");
    cat.query();
    if (cat.next()) {
      filter += " OR type!=" + cat.sys_id;
    }
  }
  gs.log("TaskAssignmentFilter: " + filter);
  return filter;
}
```

Once the business rule above is defined, it can be called by the following reference qualifier:

```javascript
javascript:TaskAssignmentFilter()
```
User self-registration

The User Registration Request plugin provides the ability for unregistered users to request access to a ServiceNow instance. An administrator can activate the plugin.

A user can request an account by navigating to the instance. If the plugin is installed, the following section is added to the welcome screen.

---

**Request a user account**

*If you do not yet have a user account you can request one using the self registration form.*

The user can complete and submit the self-registration form, and see a confirmation that it was submitted. An email is received by the user when the account is registered.

**Note:** If the email address entered in the self-registration form is already in the system, the request is not submitted.

Approve an account

When a user submits a self-registration form, it can be reviewed and approved.

Role required: admin

1. Navigate to **User Administration > Pending User Registration** and open the request.
2. Use the **Create User** and **Reject** related links on the registration request form to approve or deny the request.
   - If **Create User** is selected, a new user is created using the email address as the **User ID**.
   - If **Reject** is selected, the request is marked **Rejected**.

The user is sent an email notification with the login information if the request was accepted, or the rejection information.

3. Optional: To view past registration requests, remove the **[State] [is] [Pending]** breadcrumb from the list view.

---

Enable auto processing of a request

You can enable auto processing of user self-registration requests by setting a property.
Role required: admin

1. Navigate to **System Properties > System**.
2. Select the check box for the property Enable auto processing of user registration requests.
3. Click **Save**.
   - If enabled, registration requests do not require approval. Instead, the business rule Auto-Process User Registration creates the user record from the information provided.

### Impersonate a user

Administrators can impersonate other authenticated users for testing purposes.

When impersonating another user, the administrator has access to exactly what that user would have access to in the system, including the same menus and modules. ServiceNow records anything the administrator does while impersonating another user as having been done by that user.

Use this feature to test what different users can do in the system and to perform actions for them in their stead.

### Useful logins

Several different logins are recommended to test the system.

Several different logins are recommended to test the system:

- An admin account to do work.
- An itil (or similar) login to test as a technician.
- An ess login to test as an end user.

More logins may be required to adequately test the system.

---

**Note:** When you impersonate a user who is locked out or is inactive, the system forces you out of the system as well after you generate an event or click a link.

---

### Impersonate a user in UI16

How to impersonate a user in UI16.

Role required: impersonator

1. In the banner frame, click your user name to open the user menu.
2. Select **Impersonate User**.
   - The Impersonate User dialog box appears.
3. Select a user from the Recent Impersonations list or enter a different user's name in the user selection field.

4. To return to your original login, follow these same steps then select your name from the list.

Impersonate a user in UI15 or UI11

How to impersonate a user in UI15 or UI11.

Role required: impersonator

1. Click the impersonate icon (in UI15, in UI11) in the banner frame. The Impersonate User dialog box appears.

2. Select the user from the Recent Impersonations list, click the lookup icon and select the user's name from the full list, or type the user's name.
3. Click OK.

Impersonate a user on a mobile phone

How to impersonate a user on a mobile phone.

The impersonation option is not visible in the mobile view of the platform, and impersonating is not supported for mobile phones. For most mobile phones, however, it is possible to impersonate a user by switching to standard view, performing the impersonation, and switching back to mobile view. Some mobile devices may have problems rendering the Impersonation dialog.

Modify the impersonate option

You can modify, enable, or disable the impersonate option.

Invoke or modify the impersonate option

The impersonation option and its effects are contained in a UI macro.

The UI macro is called impersonate_button. Modifying this UI macro is not recommended.

Enable or disable the impersonate option

The impersonation capability can be enabled/disabled with the glide.ui.impersonate_button.enable UI Property, "Enable impersonation button in banner line".
Log impersonations

Impersonations are logged in the System Log.
Logging can be enabled or disabled with the glide.sys.log_impersonation property.

Figure 506: Impersonate log

Force logout

In some cases, impersonating a user might cause an issue that makes it difficult to switch back (for example, if in a test environment, the user is being presented with a broken page).
To return to the user, navigate to http://<instance name>.service-now.com/logout.do and log back in.

Manage user sessions

The ServiceNow platform provides the ability to view and terminate individual user sessions, lock out users from the system, and make users inactive..

- Terminating a specific user session effectively logs that user out of the next transaction, which is usually the next browser click. Use the terminate sessions feature when you want to perform system maintenance.
- Locking a user out of the system means the user can no longer log in or generate any actions from any email messages that the user sends to the instance. Locking out users also terminates their user sessions.
- Making a user inactive means that the user does not show up in any fields that reference active users on the User table.

Modify session timeout

The base system uses the default Apache session timeout of 30 minutes.
The application logs the user out automatically, unless the Remember Me check box in the login screen is selected. Making the interval longer can lead to the unnecessary maintenance of inactive sessions in memory. It is recommended to adjust this timeout setting to no more than a few hours, although up to 24 hours is workable.

Note: Regardless of how many windows a user has open in a browser, it is considered to be one session. However, if a user has two separate browsers open (such as Internet Explorer and Firefox), it is considered to be two separate sessions.

To set the session timeout manually:

1. Clear the Remember Me check box in the login screen.
2. Add a new property using the following values:
   - Name: glide.ui.session_timeout
• **Description:** Type a brief description. In this case, enter something like *Override the default session timeout (30). This value is in minutes.*

• **Type:** Select the appropriate data type. In this case, select **Integer**.

• **Value:** Change the default value from 30 minutes to a value of your choice.

**Note:** The session timeout can also be set through installation exit customizations.

**Notes and limitations:**

- Ajax calls to the server keep the session alive (such as Labels and Refreshing homepages).
- Polling keeps the session alive when the chat desktop is open (requires the **Chat** plugin).
- Administrators can add the following properties to the System Properties table.
  - `glide.security.csrf.handle.ajax.timeout`: Handles errors for timed out Ajax requests when set to true.
  - `glide.security.auto.resubmit.ajax`: Automatically resubmits timed-out Ajax requests when set to true and the **Remember Me** checkbox is selected or **automatically set**. A popup appears to users asking them to continue.
  - `glide.ui.auto_req.extend.session`: When set to **true**, the system automatically extends a user’s session by the value the user selects for the homepage refresh time. If there is no homepage refresh time, the standard timeout value applies. Tablet and mobile devices do not support this property. When set to **false**, user sessions will timeout if the Remember me checkbox is clear. The timeout is based on whether there is a homepage refresh time. When there is no homepage refresh time, the standard timeout value applies. When there is a homepage refresh time, the user session times out after the timeout value plus one interval of the homepage refresh time. For example, if a user selects to refresh interval of 5 minutes, then user sessions expires after the timeout value plus five minutes.
**Note:** Users who select the **Remember me** checkbox will never timeout and are unaffected by session timeout properties.

---

**Lock out a user**

Lock out a user when you do not want the user to access the instance.

1. From the left navigation pane, select **User Administration > Users**, and select the user from the list.
2. Select the **Locked Out** check box, and update the record.

---

**Mark a user inactive**

How to mark a user inactive.

Making a user inactive does not lock out the user. The Lock Out Inactive Users business rule, which is active by default in all instances, sets the **Locked Out** flag to **true** on the User record when the Active flag is set to **false**. If you do not have this business rule active, inactive users are not automatically locked out and can still log in the instance.

1. From the left navigation pane, select **User Administration > Users**, and select the user from the list.
2. Uncheck the **Active** checkbox, and update the record.
Terminate a specific user session

How to terminate a specific user session.

1. From the left navigation pane, select **User Administration > Logged in users**. You can only see users who are logged into the same application node as you. If the Active field on a user record value is false, the user is logged in but not currently running a transaction. Most users appear as inactive at any given time.

2. Select the session you want to end.

3. Click **Lock Out Session**.

   The session is terminated, and the user is redirected to the login page at the next attempted transaction. The user is not locked out. Multiple user sessions may be associated with one user; terminating a user session only affects the specific session.
System Usage

The System Usage modules track usage for ServiceNow applications and for ServiceNow Store applications.

The usage analytics process collects data on all your instances and regularly updates the reports in the Usage Overview and ServiceNow Store Usage Overview modules. Application usage data is collected whenever an application is opened, and counts on tables are collected once a day. Data is collected on:

- The number of active users in the system
- The hardware CIs discovered (for instances that use Discovery)
- The number of cloud management service catalog items available to users in instances that use Cloud Management

All users with the admin role can view Usage Overview and ServiceNow Store Usage Overview reports.

System Usage Overview reports

The Usage Overview modules display reports on usage of ServiceNow applications and ServiceNow Store applications.

See the illustration for tips on using the charts. Navigate to **System Usage** > **Usage Overview** and **System Usage** > **ServiceNow Store Usage Overview** to view the following reports and charts:
<table>
<thead>
<tr>
<th>Chart or report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage of ServiceNow Applications and Usage of ServiceNow Store Applications</td>
<td>Shows the number of accesses (views or operations performed) for each of the listed applications, grouped by month.</td>
</tr>
<tr>
<td>Users of ServiceNow Applications and Users of ServiceNow Store Applications</td>
<td>Shows the number of active users who have used the instance, grouped by month. An active user is any user who could have accessed the applications, not only users who actually did access the applications.</td>
</tr>
<tr>
<td>Additional Metrics</td>
<td>Shows any of the following items, depending on your licensing agreement:</td>
</tr>
<tr>
<td></td>
<td>• Configuration automation nodes</td>
</tr>
<tr>
<td></td>
<td>• Custom Orchestration workflow activities</td>
</tr>
<tr>
<td></td>
<td>• Hardware CIs found by Discovery</td>
</tr>
<tr>
<td></td>
<td>• Public catalogs your organization is using</td>
</tr>
<tr>
<td></td>
<td>• Password credentials that the Password Reset application is managing</td>
</tr>
<tr>
<td></td>
<td>• Performance analytics indicators</td>
</tr>
<tr>
<td>Other Metrics (ServiceNow applications only)</td>
<td>Shows the number of items that extend a table other than the Task base table or a custom table.</td>
</tr>
</tbody>
</table>
Figure 507: Working with charts on the Usage Overview report
On-call scheduling

The on-call scheduling application provides a way to determine which member of a user group is available to complete a task.

For example, finding the right person to assign an incident. It does this by rotating an on-call position within some or all members of that group of users on a regular basis.

On-call scheduling can help answer questions like:

• For a specific group, who is the primary contact person right now?
• Who is the primary contact at any given time?
• How do I escalate notifications for this group?
• When am I on-call for this group this year?

Activate on-call scheduling

Administrators can activate the On-Call Scheduling plugin.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Installed with on-call scheduling

Several types of components are installed with on-call scheduling.

Demo data is available with on-call scheduling.

Tables installed with on-call scheduling

On-call scheduling adds the following tables.

Tables

Table 413: Tables for on-call scheduling

<table>
<thead>
<tr>
<th>Display name [Table name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate Rotation Schedule [v_alternate_rotation]</td>
<td>Lists the alternate user schedules by rota and roster and the start and end date and time stamps.</td>
</tr>
<tr>
<td>On-call Member [cmn_rota_member]</td>
<td>Lists the selected group members participating in the on-call schedule and escalations.</td>
</tr>
<tr>
<td>On-call Notifications [v_on_call]</td>
<td>Lists the reminder notifications that were sent.</td>
</tr>
<tr>
<td>Display name [Table name]</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Roster [cmn_rota_roster]</td>
<td>Determines the members on-call, the rotation interval and escalation settings.</td>
</tr>
<tr>
<td>Roster Schedule Span [roster_schedule_span]</td>
<td>Contains the schedule span definitions. on-call scheduling adds a group reference and the on-call type to the Type field.</td>
</tr>
<tr>
<td>Rota [cmn_rota]</td>
<td>Holds the on-call schedule for a particular shift.</td>
</tr>
<tr>
<td>Rotation [v_rotation]</td>
<td>Lists rotation schedules by start date and time and includes user contact information, if available.</td>
</tr>
<tr>
<td>Rotation Escalation [cmn_rota_escalation]</td>
<td>Lists the escalations including the event and last updated time stamp.</td>
</tr>
<tr>
<td>Trigger Rule [trigger_rule]</td>
<td>Extends the Assignment Rule [sysrule_assignment] table and stores when the escalation process is triggered and what actions to take.</td>
</tr>
<tr>
<td>User Rotation Schedule [v_user_rotation]</td>
<td>Lists the rotation schedule by user.</td>
</tr>
</tbody>
</table>

**Note:** The tables whose name starts with prefix v such as [v_alternate_rotation] are view-only tables and are used for generating reports.

**Properties installed with on-call scheduling**

On-call scheduling adds the following properties.

**Note:** To open the System Property [sys_properties] table, enter sys_properties.list in the navigation filter.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.snc.on_call_rotation.access.debug</td>
<td>Enable logs for debugging.</td>
</tr>
<tr>
<td>• <strong>Location</strong>: System Property [sys_properties] table</td>
<td></td>
</tr>
</tbody>
</table>

**Roles installed with on-call scheduling**

On-call scheduling adds the following user roles.
## Roles

### Table 414: Roles for on-call scheduling

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>Contains roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rota administrator [rota_admin]</td>
<td>Creates new rotas and edits or deletes existing rotas. Can also manage all other aspects of on-call rotas and rosters (duty shifts).</td>
<td>• assignment_rule_admin</td>
</tr>
<tr>
<td>Rota manager [rota_manager]</td>
<td>Has delegated access to a specific group's rota and can manage this group's rota and associated data. Cannot edit or delete rotas.</td>
<td>• none</td>
</tr>
<tr>
<td>Roster administrator [roster_admin]</td>
<td>Manages on-call rotation information.</td>
<td>• none</td>
</tr>
</tbody>
</table>

### Script includes installed with on-call scheduling

On-call scheduling adds the following script includes.

<table>
<thead>
<tr>
<th>Script include</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnCallFilters</td>
<td>Reference qualifiers for on-call rota or roster lookup.</td>
</tr>
<tr>
<td>OnCallReminderEmailGenerator</td>
<td>Generates an HTML email used to send reminders to users that have on-call duty.</td>
</tr>
<tr>
<td>OnCallRemindersNG</td>
<td>This is run by a schedule job to send reminders to on-call persons on a daily basis.</td>
</tr>
<tr>
<td>OnCallRotation</td>
<td>This class wraps the On-call Rotation support to make it easy to use from a business rule or script.</td>
</tr>
<tr>
<td>SncOnCallRotation</td>
<td>Legacy wrapper to maintain old GlideScriptable calls that have been migrated to use reflection rhino prefixes.</td>
</tr>
<tr>
<td>OnCallRotationCalculator</td>
<td>Calculate the rotation for a group, storing the results in the Rotation Schedule [v_rotation] table.</td>
</tr>
<tr>
<td>OnCallRotationPersonal</td>
<td>API for quickly getting on-call rotation data relevant to current logged in user.</td>
</tr>
<tr>
<td>OnCallRotationRecalc</td>
<td>Recalculates on-call rotation schedules. Called from Update Rotation Schedules business rules on On-call Member [cmn_rota_member] and Roster [cmn_rota_roster] tables.</td>
</tr>
<tr>
<td>Script include</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OnCallSecurityNG</td>
<td>New security model checks for on-call rotas, rotations, and associated data. Access model: • Rota_admin role gives access to manipulate all aspects of rotas. • Rota_manager role gives access to a specific group’s rota and can also modify that group’s rota and associated data.</td>
</tr>
<tr>
<td>OnCallUserReminder</td>
<td>User reminder data for sending on-call user reminder emails. Used by the On-Call Reminders NG script include.</td>
</tr>
<tr>
<td>RotaScheduleEntryValidation</td>
<td>Specific validation for rota schedule entries. Called by Rota Schedule Item Validate business rule.</td>
</tr>
<tr>
<td>ValidateSchedule</td>
<td>Checks if schedule spans start and end dates are valid and that they don't overlap with any other rotation schedule.</td>
</tr>
<tr>
<td>RosterMember</td>
<td>Handles roster and member requests.</td>
</tr>
<tr>
<td>FormattedScheduleReport</td>
<td>Generates formatted report on the on-call schedule for the selected groups or the given rotation in the a time span.</td>
</tr>
</tbody>
</table>

Client scripts installed with on-call scheduling

On-call scheduling adds the following client scripts.

<table>
<thead>
<tr>
<th>Script</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle Group Visibility</td>
<td>Schedule Entry</td>
<td>Handles group visibility when creating schedule entries for on-call. [cmn_schedule_span]</td>
</tr>
<tr>
<td>Question Choice Related List</td>
<td>Question</td>
<td>Handles the choice related list when sending on-call notifications. [question]</td>
</tr>
<tr>
<td>Validate</td>
<td>Roster</td>
<td>Provides client side validation for Roster fields. [cmn_rota_roster]</td>
</tr>
</tbody>
</table>

Business rules installed with on-call scheduling

On-call scheduling adds the following business rules.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnCallEscalation</td>
<td>None</td>
<td>Default escalation handler for on-call rotation escalations.</td>
</tr>
<tr>
<td>Remove rotation records</td>
<td>[sys_user_grmember]</td>
<td>Rotation records removed for a user group member.</td>
</tr>
<tr>
<td>Business rule</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Change active</td>
<td>[cmn_rota_roster]</td>
<td>Activate or deactivate the roster.</td>
</tr>
<tr>
<td>Update Rotation Schedules (Member)</td>
<td>[cmn_rota_member]</td>
<td>Recompute the rotation schedules if a member order value was changed.</td>
</tr>
<tr>
<td>Initial Roster Members</td>
<td>[cmn_rota_roster]</td>
<td>Creates a new group member when a roster is created.</td>
</tr>
<tr>
<td>Show records for user</td>
<td>[v_on_call]</td>
<td>Show schedules for a specific user.</td>
</tr>
<tr>
<td>Rota Updated</td>
<td>[cmn_rota]</td>
<td>Recompute the rotation schedules for the roster members after the rota has been updated.</td>
</tr>
<tr>
<td>Update Rotation Schedules (Roster)</td>
<td>[cmn_rota_roster]</td>
<td>Recompute the rotation schedules when the m2m or the roster definition change.</td>
</tr>
<tr>
<td>Delete roster member schedule</td>
<td>[cmn_rota_member]</td>
<td>Recalculate the rotation when a member is removed.</td>
</tr>
<tr>
<td>Delete group member on-call schedule</td>
<td>[sys_user_grmember]</td>
<td>Delete all spans associated to the group member being deleted.</td>
</tr>
<tr>
<td>Delete Rota Schedule</td>
<td>[cmn_rota]</td>
<td>Delete the schedule when the rota is deleted.</td>
</tr>
<tr>
<td>Validate Rota</td>
<td>[cmn_rota]</td>
<td>Verify the schedule entry being updated is valid.</td>
</tr>
<tr>
<td>Edit Schedule</td>
<td>[v_rotation]</td>
<td>Redirects to the schedule page passing the group as parameter.</td>
</tr>
<tr>
<td>Show records for user</td>
<td>[v_rotation]</td>
<td>Display notification.</td>
</tr>
<tr>
<td>Rota Schedule Item validate</td>
<td>[cmn_schedule_span]</td>
<td>Validate that the schedule entry is valid</td>
</tr>
<tr>
<td>Roster Properties</td>
<td>[v_rotation]</td>
<td>Redirects to a list of rosters filtered by groups.</td>
</tr>
<tr>
<td>Refresh Report</td>
<td>[v_rotation]</td>
<td>Refreshes the UI when UI action is executed.</td>
</tr>
</tbody>
</table>

Email notifications installed with on-call scheduling

On-call scheduling adds the following email notifications.

<table>
<thead>
<tr>
<th>Email notification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-Call Reminder</td>
<td>Sends a reminder to an on-call member when their shift is upcoming.</td>
</tr>
</tbody>
</table>

Events installed with on-call scheduling
On-call scheduling adds the following events.

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rota.on_call.reminder</td>
<td>Reminder for upcoming on-call shift.</td>
</tr>
<tr>
<td>incident.on_call.user</td>
<td>On-call user for an assignment group should be notified.</td>
</tr>
<tr>
<td>incident.on_call.device</td>
<td>On-call device for an assignment group should be notified.</td>
</tr>
<tr>
<td>incident.on_call.escalation.device</td>
<td>On-call escalation to a device occurred for a particular roster.</td>
</tr>
<tr>
<td>incident.on_call.escalation.user</td>
<td>On-call escalation to a user occurred for a particular roster.</td>
</tr>
<tr>
<td>trigger_rule.process</td>
<td>Process trigger rule will process the event.</td>
</tr>
</tbody>
</table>

Scheduled jobs installed with on-call scheduling

On-call scheduling adds the following scheduled jobs.

<table>
<thead>
<tr>
<th>Schedule job</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-Call Reminders</td>
<td>Run by a schedule to send reminders to on-calls on a daily basis.</td>
</tr>
</tbody>
</table>

On-call scheduling concepts

On-call scheduling comprises a number of elements.

- **Rota**: the calendar definition of on-call shift hours, personnel lists, and escalation settings, such as escalation type and a catch-all for a group. Rotas define the time slots within which the duty schedule is active.
- **Rosters**: a list of users who are part of the schedule. Rosters define which users are assigned to which time slots within the rota.
- **Schedules**: the basic entity from which rotas and rosters are defined. For example, a company that wants coverage of tasks around the clock would use a 24-7 schedule. Companies that provide support around the globe, could use a follow the sun schedule to cover different time zones across different continents.
- **Holidays**: time off for participants in a rota can be planned and managed in the on-call calendars.
- **Escalations**: the chain of persons and the actions to be taken, for example, when a P1 incident comes in.
- **Notifications**: can be sent to remind people who are on-call of their obligations or if an important event occurs. Notifications are by email with on-call scheduling, or by voice mail or SMS with Notify.

On call scheduling roles

For more information, see User Roles.
Table 415: Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>Contains roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rota administrator</td>
<td>Creates new rotas and edits or deletes existing rotas. Can also manage all other aspects of on-call rotas and rosters (duty shifts).</td>
<td>• assignment_rule_admin</td>
</tr>
<tr>
<td>[rota_admin]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rota manager</td>
<td>Has delegated access to a specific group's rota and can manage this group's rota and associated data. Cannot edit or delete rotas.</td>
<td>• none</td>
</tr>
<tr>
<td>[rota_manager]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roster administrator</td>
<td>Manages on-call rotation information.</td>
<td>• none</td>
</tr>
<tr>
<td>[roster_admin]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Delegating the rota manager role

As a rota manager, you can delegate your role to a member of your rota.

Role required: admin or rota_manager

1. Navigate to **User Administration > Delegate Roles in Group**.
2. Follow the instructions to **delegate** the rota manager role to a member of your rota.

**Note:** Ensure that you select your rota group within which you want to delegate the rota manager role.

3. Submit the role delegation request.

On-call scheduling wizard

A simple way to set up a new on-call schedule is to use the wizard functionality.

On-call scheduling can easily be used with Notify, which can send notifications by SMS and voice mail. See **Use Notify with on-call scheduling**.

After running the on-call wizard, you can adjust individual rosters, and escalation and reminder settings to fine-tune your setup. You can also enable on-call notifications by setting up rotation workflow triggers.

Schedule templates

You can also define schedule templates that are used in the new schedule wizard. For more information, see **Use Schedules**.

Set up a new on-call schedule

You can use the on-call scheduling wizard to add a new on-call schedule.

Role required: rota_admin or admin

1. Navigate to **On-call Scheduling > Administration > Create New Schedule**.
   The first page of the wizard opens where you define the basic information for your schedule.

2. Answer the questions, and then click **Next**.
Table 416: Basic info questions

<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What would you like your schedule’s name to be?</td>
<td>The friendly name users see when they view the rotas.</td>
</tr>
<tr>
<td>Which group does this schedule apply to?</td>
<td>The group this on-call schedule affects. Only members of this group can be added to this schedule’s rosters. A group is not allowed to have overlapping schedules. For example, if a group has a 24x7 schedule, you cannot create another schedule for the same group. However, for a group with an 8 A.M. to 8 P.M. schedule, you can create an 8 P.M. to 8 A.M. schedule.</td>
</tr>
<tr>
<td>When would you like your new schedule to begin?</td>
<td>The starting date for the on-call schedule. By default, this value is set to the current date.</td>
</tr>
</tbody>
</table>

The Schedule Definition page opens.

3. Answer the questions, and then click **Next**.

Table 417: Schedule definition questions

<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to use an existing schedule?</td>
<td>Select <strong>Yes</strong>, and then select one of the predefined options in the next question. Select <strong>No</strong> to set up the schedule’s configuration manually.</td>
</tr>
<tr>
<td>What type of schedule would you like to use?</td>
<td>Some options are available in the base system, for example, 24x7, Workday 8-5. Select the value to base your schedule on. This question appears only if you answer <strong>Yes</strong> to the first question. The logged-in user’s time zone is used to build these schedules. If the logged-in user does not have a time zone specified, the system time zone is used.</td>
</tr>
</tbody>
</table>

The following questions appear if you answered No to the first question.

<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What would you like your new schedule’s name to be?</td>
<td>The friendly name of the schedule that appears on the <strong>On-call calendar</strong> page.</td>
</tr>
<tr>
<td>Is the shift for this schedule all day?</td>
<td>Option for indicating whether each shift is a 24-hour shift. If you select <strong>Yes</strong>, the shift start time is reset to 00:00:00 and the end time is no longer required.</td>
</tr>
<tr>
<td>Question</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>When must the shift for your new schedule start?</td>
<td>The time of day the shift is scheduled to start. For a 24-hour shift, this is set to 00:00:00.</td>
</tr>
<tr>
<td>When must the shift for your new schedule end?</td>
<td>The time of day the shift is scheduled to end. The start and end time represent one shift, and the date is only different if the shift spans midnight. For example, for the 8 A.M. to 8 P.M. shift, the start is 2014-01-01 08:00:00 and end is 2014-01-01 19:59:59. For the 8 P.M. to 8 A.M. shift, the start is 2014-01-01 20:00:00 and the end is 2014-01-02 07:59:59. This field is available only for shifts that are not 24-hour.</td>
</tr>
<tr>
<td>In which time zone does your schedule apply?</td>
<td>By default the logged-in user's time zone is selected. If the schedule needs to be set up in a different time zone, change the value.</td>
</tr>
<tr>
<td>How often does 1 shift repeat?</td>
<td>The days that the shift repeats, such as Daily, Weekly, or Every Weekday (Mon-Fri). Several options are listed by default. If the value needed is not in the list, select the closest match and manually edit it after the on-call schedule is generated. The repetition interval determines how long it takes to rotate each member in the schedule. For example, if you select Daily, then each member's shift in the schedule rotates on a daily basis. If you select Weekly, then each shift continues for a week before rotating to the next member in the roster. The wizard does not provide a monthly or yearly option. Although you could create a monthly or a yearly shift by editing the schedule manually, this is not advised because it is hard to determine the on-call person over a long period of time. In this case, create a yearly or a monthly schedule with only one roster and one member so that the same person is always on-call. Any exceptions can be made as a one-time change in the roster.</td>
</tr>
</tbody>
</table>
The Escalations and notifications setup page opens.

4. Answer the questions, and then click Next.

### Table 418: Escalation and notification settings

<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many duty rotations are there in your on-call schedule?</td>
<td>The simplest on-call schedule contains only one roster. Selecting more than one generates multiple rosters with the same members shifted by one position in each.</td>
</tr>
<tr>
<td>Which members would you like to be in your schedule?</td>
<td>Select from the people in the group that you entered on the Basic Information page.</td>
</tr>
<tr>
<td>How many reminders should be sent out before escalating?</td>
<td>When escalations are configured for this group, the system first sends the number of reminders designated here before notifying the backup personnel.</td>
</tr>
<tr>
<td>How long should we wait until sending the next reminder?</td>
<td>By default, there is a 15 minute wait before the next reminder is sent. If the needed value is not available, select the closest match and edit the roster to update the number of reminders after the on-call schedule is generated.</td>
</tr>
</tbody>
</table>

The following results occur.

- An on-call schedule is created.
- A roster is created for the number of duty rotations specified.
  - Each roster follows the specified schedule.
  - A lineup of the selected members from the group is created. For multiple duty rotations, the order of the members is offset by 1 to prevent scenarios where users are scheduled as their own backup during an escalation.
- Escalation settings are created.

### Change rota escalation settings

Rota escalations are set after the wizard has been completed.

Role required: rota_admin or admin

**Escalation type** determines the order in which certain members of a group are notified about the escalation. The escalation type depends on the number of rosters. Its value is automatically set to **Rotate Through Members** if a rota has only one roster. The escalation chain goes through all members of that roster. If a rota has more than one roster, its value is **Rotate Through Rosters**. The escalation chain goes through all the rosters to determine who to notify.

**Catch-all** identifies the user who receives notifications if none of the on-call staff accepted the incident assignment. It can be none, a group manager, an individual, or all roster members.

1. Navigate to **On-Call Scheduling > My Group Schedules**.
2. Select the rota to edit.
3. Change the escalation settings.
4. Click Update.

**Change rota reminder settings**
The **Send on-call reminders** and **Reminder lead time (days)** settings send an email notification to the on-call person a specified number of days before the on-call obligation.

Role required: rota_admin or admin

The reminder is sent to the email address specified in the user record, unless it is specified somewhere else in the notification preferences. Notification preferences take precedence over the user record.

1. Navigate to **On-Call Scheduling > My Group Schedules.**
2. Select the rota to edit.
3. Change the reminder settings.
4. Click **Update.**

**Access the roster for a rota**
The **Rosters** related list in the Rota form contains the defined rosters. Each roster identifies a subset of group members who participate in the on-call roster.

Role required: rota_admin or admin

1. To access rosters for a specific rota, navigate to **On-Call Scheduling > On-Call Calendars.**
2. Right-click a rota and select **Edit rota.**
   
   If a roster’s rotation should begin at a particular time of day, perhaps on-call responsibility transfers at 8 A.M. instead of midnight, clear the **All day rotation** check box so you can specify a start time.
3. Open a roster to see the members.
   
   Initially, roster members are automatically populated from the **Group.**
4. Optional: To remove users who do not participate in a rotation, click the **Edit** button on the **Members** related list, use the slushbucket to remove members, and then click **Save.**
   
   You cannot add members to a roster who are not in the group.

**Escalation and reminder settings**
Escalation settings allow you to control how and when escalations occur and notifications are sent.

**Escalation settings**

- **Forced communication channel:** If Notify is active, you can specify a mandatory communication channel, either SMS or email. Keep in mind that if the on-call member does not have an SMS device, they are not contacted. No further communication attempts are made and the lack of an SMS device is logged.
- **# reminders:** The number of times the ServiceNow platform sends reminders to a person who doesn't reply within the time frame defined in **Time between reminders.**
- **Time between reminders:** The time between the reminders being sent.

Values in these fields determine the value in the **Time before escalation** field. For example, if **# reminders** is 2 and **Time between reminders** is 10 minutes, then the **Time before escalation** is 30 minutes. That is the time that passes between the first notification of a specific person and the first notification of the next person if the first one does not reply to the notification or reminders.
Reminder settings

Reminder settings can be defined for each individual roster. They enable configuration of sending reminders and the reminder lead time in days.

On-call notifications

To enable on-call notifications so that rota workflows have an effect, you must define trigger rules.

Trigger rules determine the conditions that must be met before a notification is sent and what action must be taken. For more information, see Trigger Rules.

On-call scheduling management

You can edit a rotation schedule directly from the calendar.

After you create a rotation schedule, you can edit it directly from the calendar. This allows easy access to the rotation, especially for one-time changes. Users with the admin, rota_admin, or rota_manager role can edit schedule data. Only users with the admin or rota_admin role can delete schedule data.

Adjust an existing shift

You can adjust an existing schedule from the on-call calendar.

Role required: rota_admin or rota_manager

1. Navigate to On-Call Scheduling > On-Call Calendars.
2. Select a Group.
   The rotation schedules are shown for that group.
3. Double-click an existing shift.
4. Select one of the following options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify extra times when a rota is active</td>
<td>Select the Rota to change if multiple rotas are available. If a span overlaps with another rotation schedule, an error is shown.</td>
</tr>
<tr>
<td>Schedule time off for a group member</td>
<td>Select the Member to schedule time off for.</td>
</tr>
<tr>
<td></td>
<td>Time off is shown as a different color, and the member’s name is followed by a time off notation. To give a group member time off for more than one day, double-click to open the time-off record you just created and set it to repeat, for example, daily until a specified date.</td>
</tr>
</tbody>
</table>

Note: When you specify time off for a group member, it only applies to the current group selected. If users are on multiple rotas, time off must be entered separately for each rota.

5. Click OK.

Designate a substitute for a shift

You can substitute a shift from the on-call calendar.

Role required: rota_admin or rota_manager

1. Navigate to On-Call Scheduling > On-Call Calendars.
2. Select a Group.
The rotation schedules are shown for that group.

3. Double-click the appropriate shift.

4. Select **Provide on-call coverage for another roster member**.

5. Select the **Member** scheduled to work the shift as a substitute.

6. In **Covering on**, select the desired roster or **All**.

7. Click **OK**.

The changed shift is shown in orange and the information for the roster schedule entry states the selected shift, for example, **Primary coverage**, or **Secondary coverage**. If you selected **All**, the member you selected covers for all rosters at that particular time and as many entries are made as there are rosters.

The final result looks like the following illustration.
Change the rota or span for a group
You can change the rota or span for a group from the on-call calendar.

Role required: rota_admin or rota_manager

Note: When creating or changing calendar entries manually via On-Call Scheduling > On-Call Calendars, enter the start time as a whole hour and the end time minus one second to prevent
unintentional overlap with other entries, resulting in an incorrect on-call lineup. For example, a span from 16:00 to 24:00 P.M. should be entered as: 16:00:00 to 23:59:59.

1. Navigate to On-Call Scheduling > On-Call Calendars.
2. Right-click the appropriate rota and select one of the following options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit rota</td>
<td>To change the schedule or related information.</td>
</tr>
<tr>
<td>Edit span</td>
<td>To change the dates and frequency that the rota is active.</td>
</tr>
</tbody>
</table>

3. Make the changes and click Submit.
   If you click Resend Reminders on the Rota form, you can inform the rota members of the changes you made.

On-call scheduling ITIL functions

Users with the itil role can use on-call scheduling functions to keep track of rotation schedules.

The following list describes these ITIL functions.

- View schedules for their groups.
  - View their personal schedule.
- View calendars on page 1785
- View reports

View my group schedules
You can see the on-call schedules for groups that you are a member of.

Role required: itil

1. Navigate to On-Call Scheduling > My Group Schedules.
2. Open a schedule to view the details.
3. Optional: To view a roster, click its order number in the Rosters related list.

View my schedule
You can view when you are on-call and the escalation lineup for a particular date range.

Role required: itil

1. Navigate to On-Call Scheduling > My Schedule Report.
2. In Show for, select a date range.
3. Click Submit.
This shows the rota, the roster, and the start and end times for each timeslot.

4. To view alternate rosters, expand an entry by clicking the arrow.
This shows other rosters for that timeslot. For example, if you are the primary contact, this shows secondary and tertiary contacts for that timeslot.

**View calendars**
On-call calendars provide a way of visualizing the on-call rotation for a group.

*Role required: itil*

A calendar shows the rotation schedule in light blue and the roster in deep blue for a specific rotation. If there is more than one rotation schedule on a particular day, the secondary rotation is shown in green and the tertiary in yellow.
1. Navigate to **On-Call Scheduling > On-Call Calendars**. The calendar display defaults to the first group with a roster. Each time slot specified for the group’s roster is displayed along with the on-call person assigned to that slot.

2. In the **Group** choice list, select the group you are interested in viewing.

3. Use any of the following icons and controls to navigate within the calendar.

---

### Figure 508: On-Call Calendar

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00</td>
<td>00:00</td>
<td>00:00</td>
<td>00:00</td>
<td>00:00</td>
<td>00:00</td>
<td>00:00</td>
</tr>
<tr>
<td>01:00</td>
<td>01:00</td>
<td>01:00</td>
<td>01:00</td>
<td>01:00</td>
<td>01:00</td>
<td>01:00</td>
</tr>
<tr>
<td>02:00</td>
<td>02:00</td>
<td>02:00</td>
<td>02:00</td>
<td>02:00</td>
<td>02:00</td>
<td>02:00</td>
</tr>
<tr>
<td>03:00</td>
<td>03:00</td>
<td>03:00</td>
<td>03:00</td>
<td>03:00</td>
<td>03:00</td>
<td>03:00</td>
</tr>
<tr>
<td>04:00</td>
<td>04:00</td>
<td>04:00</td>
<td>04:00</td>
<td>04:00</td>
<td>04:00</td>
<td>04:00</td>
</tr>
<tr>
<td>05:00</td>
<td>05:00</td>
<td>05:00</td>
<td>05:00</td>
<td>05:00</td>
<td>05:00</td>
<td>05:00</td>
</tr>
<tr>
<td>06:00</td>
<td>06:00</td>
<td>06:00</td>
<td>06:00</td>
<td>06:00</td>
<td>06:00</td>
<td>06:00</td>
</tr>
<tr>
<td>07:00</td>
<td>07:00</td>
<td>07:00</td>
<td>07:00</td>
<td>07:00</td>
<td>07:00</td>
<td>07:00</td>
</tr>
<tr>
<td>08:00</td>
<td>08:00</td>
<td>08:00</td>
<td>08:00</td>
<td>08:00</td>
<td>08:00</td>
<td>08:00</td>
</tr>
<tr>
<td>09:00</td>
<td>09:00</td>
<td>09:00</td>
<td>09:00</td>
<td>09:00</td>
<td>09:00</td>
<td>09:00</td>
</tr>
<tr>
<td>10:00</td>
<td>10:00</td>
<td>10:00</td>
<td>10:00</td>
<td>10:00</td>
<td>10:00</td>
<td>10:00</td>
</tr>
<tr>
<td>11:00</td>
<td>11:00</td>
<td>11:00</td>
<td>11:00</td>
<td>11:00</td>
<td>11:00</td>
<td>11:00</td>
</tr>
<tr>
<td>12:00</td>
<td>12:00</td>
<td>12:00</td>
<td>12:00</td>
<td>12:00</td>
<td>12:00</td>
<td>12:00</td>
</tr>
<tr>
<td>13:00</td>
<td>13:00</td>
<td>13:00</td>
<td>13:00</td>
<td>13:00</td>
<td>13:00</td>
<td>13:00</td>
</tr>
<tr>
<td>14:00</td>
<td>14:00</td>
<td>14:00</td>
<td>14:00</td>
<td>14:00</td>
<td>14:00</td>
<td>14:00</td>
</tr>
<tr>
<td>15:00</td>
<td>15:00</td>
<td>15:00</td>
<td>15:00</td>
<td>15:00</td>
<td>15:00</td>
<td>15:00</td>
</tr>
<tr>
<td>16:00</td>
<td>16:00</td>
<td>16:00</td>
<td>16:00</td>
<td>16:00</td>
<td>16:00</td>
<td>16:00</td>
</tr>
<tr>
<td>17:00</td>
<td>17:00</td>
<td>17:00</td>
<td>17:00</td>
<td>17:00</td>
<td>17:00</td>
<td>17:00</td>
</tr>
<tr>
<td>18:00</td>
<td>18:00</td>
<td>18:00</td>
<td>18:00</td>
<td>18:00</td>
<td>18:00</td>
<td>18:00</td>
</tr>
<tr>
<td>19:00</td>
<td>19:00</td>
<td>19:00</td>
<td>19:00</td>
<td>19:00</td>
<td>19:00</td>
<td>19:00</td>
</tr>
</tbody>
</table>
### View the monthly calendar
Click the monthly icon ( )

### View the weekly calendar
Click the weekly icon ( )

### View the daily calendar or move to a specific date
Click the daily icon ( )

### Move the calendar back and forward in time
Click the left and right arrows beside the date.

### View the calendar as a timeline
Click the timeline icon ( )
The on-call shifts are shown as horizontal bars for each day. By default one week is shown. Use the + and - buttons to expand or collapse the timeline to 21, 35, 49, 63 days and so on.

### Add item
You can add extra coverage, specify extra times for when a rota is active, and schedule time off.
Role required: group_manager or rota_manager

1. Navigate to **On-Call Scheduling > On-Call Calendars**. The on-call coverage for the selected group’s calendar displays.
2. Double-click the time-slot for which to add a schedule entry. The **Add Item** pop-up window displays.
3. Select one of the following options, as appropriate.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Provide on-call coverage for another roster member** | Select details of the member who will provide coverage for another roster member. The following options appear when you select this option:  
• Member: Select the member to provide extra coverage  
• Covering on: Select the roster for which the member will provide coverage. |
| **Specify extra times when a rota is active** | Specify extra coverage for the selected rotation.                           |
| **Schedule time off for a group member**    | Specify the group member who will take scheduled time-off during the selected time frame. |

### Modify schedule entry
You can make updates to a schedule entry such as changing schedule entry type, date, and time.
Role required: group_manager or rota_manager

1. Navigate to **On-Call Scheduling > On-Call Calendars**. The on-call coverage for your group’s calendar displays.
2. Double-click the schedule entry on the calendar to modify.
3. Update in the fields on the pop-up form.

Table 419: Schedule Entry

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Modify the name of the schedule entry, if required.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the type of schedule entry. For example, on-call or time off.</td>
</tr>
<tr>
<td>Group</td>
<td>The <strong>Group</strong> field is left empty by design.</td>
</tr>
<tr>
<td></td>
<td><strong>Caution</strong>: If the <strong>Group</strong> field is populated, then the spans for that</td>
</tr>
<tr>
<td></td>
<td>entry are not displayed.</td>
</tr>
<tr>
<td>Show as</td>
<td>Select what the schedule entry must be displayed as. For example, busy or</td>
</tr>
<tr>
<td></td>
<td>on-call.</td>
</tr>
<tr>
<td>When</td>
<td>Select the appropriate start and end date, and time for the schedule entry.</td>
</tr>
<tr>
<td>All day check box</td>
<td>Select the check box to make the schedule entry active for the entire</td>
</tr>
<tr>
<td>Timezone</td>
<td>Displays the time zone for the schedule entry. This cannot be modified.</td>
</tr>
<tr>
<td>Repeats</td>
<td>Select the frequency with which the schedule entry repeats. For example,</td>
</tr>
<tr>
<td></td>
<td>daily or weekly.</td>
</tr>
<tr>
<td></td>
<td>The following fields appear if you select <strong>Monthly</strong>:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Monthly type</strong>: specify the date of the month when the schedule entry</td>
</tr>
<tr>
<td></td>
<td>repeats.</td>
</tr>
<tr>
<td></td>
<td>The following fields appear if you select <strong>Yearly</strong>:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Yearly type</strong>: specify the date of the year when the schedule entry</td>
</tr>
<tr>
<td></td>
<td>repeats or if it is floating. If you select the <strong>Floating</strong> option,</td>
</tr>
<tr>
<td></td>
<td>then the following fields appear:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Float week</strong>: Select the week of the month when the schedule entry</td>
</tr>
<tr>
<td></td>
<td>repeats.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Float day</strong>: Select the day of that week when the schedule entry</td>
</tr>
<tr>
<td></td>
<td>repeats.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Month</strong>: Select the month when the schedule entry repeats.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Repeat every</td>
<td>If the schedule entry is selected to repeat, then specify how often it repeats. For example, if you select weekly repetitions, specify the frequency such as every week or every two weeks.</td>
</tr>
<tr>
<td>Repeat on</td>
<td>If the schedule entry is selected to repeat, then specify when it repeats. For example, if you select weekly repetitions, specify the days of the week when it would repeat.</td>
</tr>
<tr>
<td>Repeat until</td>
<td>If the schedule entry is scheduled to repeat, then specify an end date until which the schedule entry repeats itself.</td>
</tr>
</tbody>
</table>

4. Click **Update** to save changes.

**View a report for on-call scheduling**

You can schedule a report of users who are on-call or a catch-all for one or more groups during a specific time period.

Role required: itil

1. Navigate to **On-Call Scheduling > Schedule Report**.
2. Select the start and end dates for the report.
3. Enter the first few letters of the desired group into the **Name** field to see a list of groups that start with those letters, or select the **All groups** check box to see a list of all available groups.
4. Select at least one group in the slushbucket.
5. For **Report style**, select one of the following.
   - **Table**: to display the report on-screen as a list that can be sorted, filtered, and configured, like other lists.
   - **Formatted**: to generate a report in PDF format.
6. Click **Run Report**.
7. If you selected **Formatted**, click the **Click to Print** button to print the report.
   
   The report shows the on-call commitments for all selected groups during the selected date range.
View an escalation report

The escalations report shows you what the escalation sequence is and which rules apply for the selected date. For each group, you can choose to show either the active roster members or the on-call person.

Role required: itil

You can see the on-call persons only for the groups you are authorized to see. If you select an unauthorized group, a message appears stating that a number of rows have been removed due to security constraints.

1. Navigate to On-Call Scheduling > Escalations > Escalations Report.
2. Select the date for the report.
3. Enter the first few letters of the desired group into the Name field to see a list of groups that start with those letters, or select the All groups check box to see a list of all available groups.
4. Select at least one group in the slushbucket.
5. For **Report style**, select one of the following.
   - **Active roster members**: to display the on-call persons and catch-all persons in the escalation order, along with their delay times.
   - **On call person**: to quickly find out who is currently on duty.

6. **Click Run Report**.

**Define lead time for email reminders**

On-call scheduling includes a scheduled job that checks if any on-call members need to be notified about upcoming on-call commitments. The lead time for when reminders are sent can be modified.

**Role required**: rota_admin or rota_manager

The On-Call Reminders job runs the OnCallRemindersNG script include to generate the notifications.

1. Navigate to **On-Call Scheduling > On-Call Calendars**.
2. Right-click the rota and select **Edit rota**.
   
   The rota is the listing in the calendar above the scheduled users.
3. Change the **Reminder lead time (days)** field for the on-call schedule record or for any of its rosters.

   The reminder lead time defined on a roster is always respected. If no lead time is defined, the on-call schedule's reminder lead time is used. If the reminder lead time is not defined for either the on-call schedule or its rosters, then a default of 2 days is used.

**Note**: Keep in mind that the Reminder lead time on the Roster form is different from the # reminders and Time between reminders fields in the Escalation Settings section of the form. The escalation settings are only used to configure reminders for escalations. The Reminder lead time is in the Reminder Communication section of the Roster form, and is used to email reminders for upcoming on-call commitments.

**Resend on-call email reminders**

You can resend email reminders, which can be useful to inform group members about changes in their shift.

**Role required**: rota_admin or admin

1. Navigate to **On-Call Scheduling > My Group Schedules**.
2. Open the rota to resend email reminders.
3. **Click Resend reminders**.
   
   The option is also available when you edit a rota for a group in **On-Call Calendars**.

**On-call scheduling escalations**

Escalations provide a mechanism to ensure that important issues are addressed in a timely manner within on-call scheduling.

Escalations use a mechanism similar to SLAs to monitor response time and take time measurements. You can take action if these times are breached. The actions, like sending out an email or an SMS, are fired by trigger rules. You define trigger rules to determine the conditions under which specified actions are taken.

For example, a critical incident is raised for Acme Pharmaceuticals regarding a problem with their network access. An SMS notification is sent to James Jones, the third-line support engineer who is currently on-call for this type of incident. However, James is unavailable, and does not respond within the defined 30-minute response time.
Trigger rules defined for Acme Pharmaceuticals' critical incidents initiate an escalation after 30 minutes without a response. The person defined as the next point-of-contact is Ken Kramer, James' line manager, so an SMS notification is sent to Ken.

Users with the rota_admin role can configure escalation settings and define trigger rules.

Escalation triggers
Escalation triggers define the conditions under which escalation actions occur in on-call scheduling. These actions can be defined with a workflow or server-side JavaScript.

When a task is created or updated, the system compares the assignment rules with the trigger conditions, to see if any of the trigger conditions are matched. If a match is found, the system activates the associated workflow or script, which holds the escalation steps and actions.

For example, a new P1 incident is created and assigned to the Software Group. A trigger condition states that if P1 incidents are created and assigned to this group, then an associated script or workflow should run.

Escalation trigger rules

Escalations use trigger rules to define conditions under which a trigger action is to be executed. A trigger action can be either a script or workflow. Trigger rules are an extension of assignment rules and are therefore run by the system. The behavior of assignment rules is used for trigger rules.

Trigger rules and trigger actions allow users to quickly set up escalation scenarios.

Example escalation scenario

Acme Pharmaceuticals needs to support a simple escalation process. When a critical or high incident is raised, a member of the Network group should be assigned to the incident based on the Network group's on-call schedule.

First, the trigger rule is defined with its conditions.
Create a trigger rule

You can create trigger rules to define conditions under which a trigger action is to be executed.

Role required: rota_admin or admin

Trigger rules fire only if the \texttt{assigned\_to} and \texttt{assignment\_group} fields are not populated on a record.

1. Navigate to \textbf{On-Call Rotations} \textgreater{} \textbf{Escalations} \textgreater{} \textbf{Trigger Rules}.
2. Click \textbf{New}.
3. Complete the form.

\begin{table}
\centering
\caption{Trigger rule form}
\begin{tabular}{|l|l|}
\hline
\textbf{Field} & \textbf{Description} \\
\hline
Name & The name of this trigger rule. \\
Order & The execution order of this trigger rule. \\
Table & A task table that applies to the trigger rule. \\
\hline
\end{tabular}
\end{table}

\textbf{Note}: The list shows only tables and database views that are in the same scope as the trigger rule.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match conditions</td>
<td>How the conditions should be applied. If you select <strong>All</strong>, all of the conditions have to be matched. If you select <strong>Any</strong>, it is sufficient if any one of the conditions is matched.</td>
</tr>
<tr>
<td>Conditions</td>
<td>The conditions which must be met before the trigger rule executes.</td>
</tr>
<tr>
<td>Group</td>
<td>Define the group that is applied to the <strong>Tasks assignment group</strong> field when the trigger rule is applied to the task.</td>
</tr>
<tr>
<td>Trigger action</td>
<td>Run a <strong>Workflow</strong> or <strong>Script</strong> when the trigger conditions are met.</td>
</tr>
<tr>
<td>Trigger workflow</td>
<td>The workflow to execute. This field is available only if <strong>Trigger action</strong> is set to <strong>Workflow</strong>.</td>
</tr>
<tr>
<td>Trigger script</td>
<td>The script to execute. This field is available only if the <strong>Trigger action</strong> is set to <strong>Script</strong>.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

**Escalations by email workflow for on-call scheduling**

The workflow enables sending an email notification regarding a newly raised incident to the on-call members of a particular group.

The workflow does not expect a response to the email notifications it sent, but instead checks the source incident record for changes in the **assigned_to** field. Depending on the value of the **assigned_to** field of the source record, the workflow branches or loops until the escalation chain is exhausted or the **assigned_to** field of the source incident is populated. If none of the users accepts assignment in time, the catch-all person is notified, if configured.

The workflow respects time-off as specified in the rosters. People who have time-off are not included in the escalation chain and no notifications are sent to them.

The workflow is intended to be used on a trigger rule based on the Incident table but other tables can be used as well. If you want to use a different table for this workflow, select the appropriate table, for example, Problem, in the **Trigger Workflow** field on the trigger rule form. This workflow is provided with on-call scheduling.
Escalation chain
An escalation chain describes the order in which rosters and roster members receive escalation notifications.

Depending on the number of rosters, the rota escalation type is one of the following.

- **Rotate through members**: the escalation chain goes through the member list of a specific roster (primary, secondary, tertiary) to determine who is to be notified.
- **Rotate through rosters**: the escalation chain goes through all the rosters to determine who is to be notified.

Depending on the escalation type you chose, the Escalations Report result can vary greatly. Following are example reports for each choice.
Figure 511: Rotate through rosters

The image shows the case in which reminders are defined for the **Primary**, **Secondary**, and **Tertiary** rosters. The members of the primary roster are notified first, then the members of the secondary, and so on for as many rosters as there are.
For this example, users can see which rotas they are part of in the My Schedule Report module. Users who are not the first on-call person are displayed as well. This enables the user to view the rota from a personal perspective, which is the preferred method for most users.

Use Notify with on-call scheduling

Within on-call scheduling, you can use Notify functions to send an SMS to on-call resources when an incident gets assigned to them.

A mobile number is required in their user record. Be aware that some phone numbers have only voice mail capabilities.

Notify is available as a separate subscription from the ServiceNow platform. To purchase a subscription, contact your ServiceNow account manager.

Notify workflow for on-call scheduling

The On-Call: Assign by Acknowledgement workflow is provided with Notify. The workflow uses data from the escalation settings of rotas and rosters. Depending on these settings, the workflow iterates through the defined escalation chain and sends notifications by SMS or email to users about incident assignment.

If Force communication channel is specified in the Escalation settings for rosters, the preferred user device is used, either SMS or email. The setting Force communication channel is only available if Notify is installed.
If the preferred device is SMS, and the on-call member does not have an SMS device defined, the user is
not contacted even if they have an email address. When forcing a communication channel on an escalation
level does not succeed, no further communication attempts are made. The fact that the user could not be
reached is logged in the log files.

The workflow respects time-off as specified in the rosters. People who have time-off are not included in the
escalation chain and no notifications are sent to them.

Before you can send notifications, you must define trigger rules. Trigger rules determine the conditions that
must be met before a notification is sent and what action must be taken. Keep in mind that trigger rules
supersede some of the on-call business rules in previous versions.

Note: If you have customized on-call business rules in previous versions, you need to deactivate
these business rules before you can work with trigger rules.

Upgrade to on-call scheduling

Group on-call rotation is replaced with on-call scheduling.

Upgrading from a previous version is completely automatic, and all events are recorded in the upgrade
logs (System Diagnostics > Upgrade History).

- The existing On-Call plugin has been changed. When you upgrade, the plugin changes are applied
  automatically (this is not optional).
- The group device functionality is deprecated in favor of a Catch All person.
- On-call scheduling replaces the existing business rules for escalations based on
  Graphical Workflow.
- The workflow uses Notification Activities, so it sends emails and not SMS messages. It must be
  modified to use Create Events activities to send SMS messages.

Keep these and the following changes in mind as you transition to on-call scheduling.

Additional workflows for Notify

Workflows have superseded the notification rules on rotation schedules.

Some example workflows are provided in the demo data available with the plugin. When you upgrade an
instance running group on-call rotation, some data from the notification rules is migrated to other tables.
The migrated information is recorded in the Notification rules - migration report, which the administrator can
run in Reports > View/Run.

Please be aware that in on-call scheduling some business rules related to escalations and notifications
were deprecated and replaced with workflows.

When you use Notify with on-call scheduling, additional workflows enable users to receive notifications and
accept or reject auto-assignment via SMS.

On-call wizard

This wizard enables users to create a basic rotation schedule and one or more rosters, along with
escalation and reminder settings.

The Create New Schedule module presents an enhanced wizard for creating schedules. The Create New
Rota module and the option for manually creating rotas are deprecated.

Option rotate through rosters

The Rotate through rosters option has been added to the My Group Schedules module.

Navigate to On-Call Scheduling > My Group Schedules and choose a rota to view this setting. It is part
of the escalation settings. The option is used automatically when there is more than one active roster. It
helps to prevent on-call users from being their own backups, as the on-call lineup order is staggered by
one for each roster.
Trigger rules upgrade
When an instance running group on-call rotation is upgraded, some data from the notification rules is migrated to other tables.

The migrated information is recorded in the Notification rules - migration report, which the administrator can run in Reports > View/Run.

Figure 513: Notification rules migration report

Trigger rules supersede the On-call Rotation Notify and On-call Rotation Cancel on-call business rules used in previous releases. If you customized on-call business rules in previous versions, you must deactivate these business rules before you can work with trigger rules.

Note: This is a one-time upgrade. If you have already upgraded trigger rules in the earlier release, you do not need to upgrade again. When you upgrade to an instance that has on-call scheduling application installed, the behavior of trigger rules for on-call scheduling changes.

Skills Management

The Skills Management feature enables an administrator to assign configured competencies, called skills, to groups or individual users. These skills can then be used to determine who can be assigned to particular tasks.
Skills can contain other skills. Any access granted to a parent skill is granted to any skill that it contains. Once a skill is assigned to a group, all members of the group automatically inherit that skill and any others contained within it. The skills mechanism is similar to the ServiceNow platform role management.

Activate skills management

The Skills Management plugin is automatically activated by the Field Service Management and Project Management plugins. Administrators can also activate the Skills Management plugin manually.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Create a skill

You can create skills to specify the competencies of your users.

Role required: skill_admin or admin

1. Navigate to Skills > Skills.
2. Click New.
3. Enter a unique, descriptive Name.
4. Enter a Description of the skill.
5. Right-click the form header and click Save.
   - The Contains Skills, Users, and Models related lists appear.
6. Optional: Use the Contains Skills related list to add subskills.
7. Optional: Use the Models related list to add any models that are associated with the skill.
   - The Users related list contains users (based on their User record or the groups they belong to) who have the skill and subskills named in this record. You can also add users to specify they have the skill.

Assign a user skill

You can assign skills individually to users.

Role required: skill_admin or admin

If you assign a skill that contains other skills to a user, the user automatically inherits the contained skills.

1. Navigate to Skills > Users.
2. Select a user from the list.
3. In the User record, select the Skills related list.
4. Click Edit and select one or more existing skills from the slushbucket.
5. Click Save.

Assign a group skill

You can assign skills to groups, and the members of the group inherit all the assigned skills.

Role required: skill_admin or admin
If you assign a skill that contains other skills to a group, the group and all its members automatically inherit the contained skills.

1. Navigate to Skills > Groups.
2. Select a group from the list.
3. In the Group record, select the Skills related list.
4. Click Edit and select one or more existing skills from the slushbucket.
5. Click Save.
   The skill is added to the group and all the group members who are granted this skill are listed at the top of the form.

Filter potential assignees based on skills

In the base system, field service management tasks and project tasks use skills to filter assignments. If a skill is identified in the Skill field, only groups or users with the appropriate skill can be assigned to the task.

The Skills Management feature contains a script include that builds a qualifier based on the assignment group and required skills for the task. For example, the Assigned To field on the Project Task record uses the following reference qualifier (using a dictionary override):

```
javascript:var util = new SkillsUtils(); util.assignedToRefQual(current);
```

This script has the following results.

- If an Assignment group is set, the list is filtered on members of that group.
- If Skills are set (the Skills field may need to be added to the form), the list is filtered on users with all the selected skills.
- If Assignment group and Skills are both set, the list is filtered on group members with the defined skills.

You can introduce the same behavior to other task tables by using the same reference qualifier.

Authentication

Authentication means validating the identify of a user who is trying to access an instance, and then authorizing the user to access features that match the user's role or job function.

Available authentication methods

You can use several different methods to authenticate users. User credentials are matched to different saved credentials for each method.

<table>
<thead>
<tr>
<th>Authentication methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local database</td>
<td>The user name and password in their user record in the instance database.</td>
</tr>
<tr>
<td>Multifactor</td>
<td>The user name and password in the database and a passcode sent to the user's mobile device that has Google Authenticator installed. See Multifactor authentication on page 1802.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Authentication methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>The user name and password in their LDAP account, which has a matching user account in the database. See <em>LDAP integration</em> on page 1809.</td>
</tr>
<tr>
<td>SAML</td>
<td>The user name and password configured in a SAML identity provider account, which has a matching user account in the database. See <em>SAML 2.0</em> on page 1904.</td>
</tr>
<tr>
<td>OAuth 2.0</td>
<td>The user name and password of OAuth identity provider, which has a matching user account in the database. See <em>OAuth 2.0</em> on page 1952.</td>
</tr>
<tr>
<td>Digest Token</td>
<td>An encrypted digest of the user name and password in the user record. See <em>Digest token authentication</em> on page 1947.</td>
</tr>
</tbody>
</table>

Multiple Provider SSO allows you to choose use several identity providers (IdPs) to manage authentication as well as retain local database authentication. You can use SAML and Digest Authentication through the Multiple Provider SSO application.

**Tips for choosing an authentication method**

- Development environments can use the instance's database credentials to speed up the development of new features.
- Customers who manage their users with existing LDAP or identity providers should use a matching authentication method.

**Enable external authentication**

You can enable external authentication on your instance.

1. Navigate to **System Definition > Installation Exits**.
2. Activate the **ExternalAuthentication** installation exit.

   **Note:** If you are using the Digest authentication method, you must also enable the DigestSingleSignOn installation exit.

3. Navigate to **System Properties > Single Sign-on**.
4. Select **Enable external authentication** and click **Save**.

Administrators may need to bypass external authentication when testing an SSO integration. Administrators can use the following URL to bypass external authentication and log in with a local user. A logged-in user cannot access this page. Attempting to access this page while logged in produces a page not found error. [http://<your-instance>.service-now.com/side_door.do](http://<your-instance>.service-now.com/side_door.do)

**Multifactor authentication**

Multifactor authentication, also known as two-step verification, is a security requirement that asserts a user enter more than one set of credentials to authenticate to an instance.
The basic level of authentication to an instance is local database authentication—the user enters a username and password combination. Multifactor authentication, in contrast, gives administrators and users the ability to require a second level of authentication—the user must enter a passcode or token in addition to the password. A mobile application on a user mobile device generates the passcode.

- Users can require multifactor authentication for their own login credentials.
- Administrators can require multifactor authentication for any user login credentials.

Multifactor authentication supports only the Google Authentication mechanism as the token provider. Users should install the recommended *Google Authenticator application* to their mobile devices.

**Supported authentication methods**

- You can use multifactor authentication in combination with the following authentication methods:
  - Local Database Authentication (native ServiceNow authentication)
  - SSO with the *LDAP integration* on page 1809

**Authentication methods that are not supported**

- Multifactor authentication is not supported in combination with the following authentication methods:
  - SSO SAML
  - SSO Digest

**Authentication flow**

*Note:* If a user is required to perform a password change while multifactor authentication is enabled on the user profile, the user does not need to enter the authorization code.

1. The user or administrator goes to a user profile in the instances and initiates multifactor authentication.
2. The instance displays a QR code and a QC code number.
3. The user takes a photo of the code with the Google Authenticator application on their mobile device, or manually enters the QC code number in the authenticator application.
4. A passcode is sent to the user's mobile device.
5. The user enters the passcode to enable multifactor authentication.
6. The next time the user tries to log in, the user looks at the Google Authenticator application to get the latest passcode.
7. The user enters the username and password and appends the passcode to the password.
8. If the username and password + passcode combination are correct, the user is authenticated to the instance.

**Activate multifactor authenticator**

Administrators can activate the *Integration- Multifactor Integration plugin*, which is not active by default.
Configure multifactor authentication

You can enable multifactor authentication on the instance and specify how many times users can skip the additional passcode requirement.

Role required: admin

1. Navigate to **Multifactor Authentication > Properties**.
2. Configure these properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Multifactor Authentication (glide.authenticate.multifactor)</td>
<td>Select this check box to allow users and administrators to use this feature.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Number of times a user can bypass multifactor authentication (glide.authenticate.multifactor)</td>
<td>Enter a number that represents how many times a user can choose to skip the additional passcode requirement. This gives your users the ability to still log in the instance if they do not have their mobile device with them. If you disable this feature and then re-enable it, the counter starts over again.</td>
<td>3</td>
</tr>
<tr>
<td>The time in minutes, the one-time code sent to user’s email address is valid for (glide.multifactor.onetime.code.validity)</td>
<td>Enter a number in minutes that specifies how long the reset code is valid. See <strong>Log on with multifactor authentication</strong> on page 1807.</td>
<td>10</td>
</tr>
</tbody>
</table>
Additional time in seconds for which the code will be valid to accommodate for the clock skew. Max value is 60 seconds. (glide.authenticate.multifactor.clock_skew)

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional time in seconds for which the code will be valid to accommodate for the clock skew. Max value is 60 seconds. (glide.authenticate.multifactor.clock_skew)</td>
<td>Enter a number in seconds with a maximum of 60. By default, the instance validates the code entered by the user against the single app-generated code generated at whatever the current time is. You can skew the time window with this property and allow one or more codes generated during a time window to be considered valid. The property's value is used in the following calculation: <code>current time - x/2</code> and <code>current time + x/2</code>, where 'x' is the value of this property. If you use the value of 10, for example, the instance considers any codes generated by the app between the time range `[the current time - 5 seconds] and [current time + 5 seconds] to be valid. Use this property to prevent log in issues where the user is unable to enter the correct code in the default time allotted.</td>
<td>10</td>
</tr>
</tbody>
</table>

3. Click Save.

### Require multifactor authentication for a user

You can require multifactor authentication for any user record in the system that you have access to.

Role required: admin

1. Navigate to User Administration > Users.
2. Configure the list to show the Enable Multifactor Authentication column.
3. Change the values of the Enable Multifactor Authentication column for the selected users to true.

When the user logs in with their user name and password, they are prompted to set up multifactor authentication.

### Set up multifactor authentication upon initial login

If your administrator enabled multifactor authentication on your profile but you have not yet set up the application, you can set it up upon login.
Role required: none

1. Log into your instance using your user name and password. The multifactor authentication setup screen intercepts your login.

2. If you want to skip the authentication setup at this time, click Bypass Setup. You can bypass multifactor authentication for a limited number of times that your administrator allows. Eventually you need to configure multifactor authentication.

3. If you do not yet have the Google Authenticator application in your mobile device, you can download it from one of the links provided under Download the app.

4. Take a photo of the QC code with the Google Authenticator application or enter the string in the application manually.

5. Enter the code and click Pair device and log in.

The application responds with a six digit code that refreshes every 30 seconds. If you entered the correct code, a message appears telling you that Multifactor Authentication is set up. If you entered
an invalid code, look for the code on your mobile device again as it may have refreshed, and enter the code that you see.

**Note:** For the code to work property, the system time on your computer must be in the same timezone as the time on your mobile device.

Log on with multifactor authentication

After multifactor authentication is enabled for your User profile, you can log in with the addition of the passcode that the Google Authenticator app gives you.

You must have multifactor authentication enabled for your profile. You can enable it yourself on your user profile or your administrator can enabled it for you.

Role required: none

1. Go to the URL of your instance to open the log in screen.
2. Open the Google Authenticator application on your mobile device and make note of the number. The number refreshes every 30 seconds.
3. Enter your user name and password, and append the passcode that you currently see on your mobile device to your password without any extra spaces. For example, if your password is **12345** and the current passcode is **424 058**, enter **12345424058**.
   
   If you forget to enter the code and you only enter the password, you are prompted to enter the code in a new window.

4. Enter the code and click **Login**.
5. If you do not remember the passcode, click the **Click to send one-time passcode** link.
   
   The temporary passcode is sent to your email address. You can append this passcode to your password as you would the original passcode. You can only use this temporary passcode once and it is only valid for 10 minutes.

**Note:** You must have an email address configured in your user profile on the instance to receive this email message.

Set up multifactor authentication on your own profile

You can set up multifactor authentication on your own User profile.
Role required: none

1. Navigate to **Self-Service > My Profile.**
2. Click the **Multifactor Authentication** related link on your User record.
   
The multifactor authentication window appears.

### Complete the steps below to enable multifactor authentication

<table>
<thead>
<tr>
<th>1. Download the app</th>
<th>2. Scan the QR Code:</th>
<th>3. Type in the App’s response:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Apple iTunes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Google Play</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Windows Store</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Or type in: KW2YUUCQ6SIBOEN

3. If you do not yet have the Google Authenticator application in your mobile device, you can download it from one of the links provided under **Download the app.**

4. Take a photo of the QC code with the Google Authenticator application or enter the string in the application manually.

5. Enter the code and click **Validate Response.**

   The application responds with a six digit code that refreshes every 30 seconds. If you entered the correct code, a message appears telling you that Multifactor Authentication is set up. If you entered an invalid code, look for the code on your mobile device again as it may have refreshed, and enter the code that you see.

**Note:** For the code to work properly, the system time on your computer must be in the same timezone as the time on your mobile device.
Disable multifactor authentication on your User profile

Users can disable multifactor authentication on their own user profile.

Role required: admin

1. Log in the instance using your multifactor authentication code appended to your password.
2. Navigate to Self-Service > My Profile.
3. Click the Multifactor Authentication related link on your User record.
4. Click Disable Multifactor Authentication.
5. Confirm the action.

You return to the user record. The setting is automatically saved. You do not need to save your User record again.

Disable multifactor authentication on the User table

You can disable multifactor authentication for the users that you have access to on the User table.

Role required: admin

1. Log in the instance.
2. Open the User [sys_user] table.
3. Configure the list to show the Enable Multifactor Authentication column.
4. Change the values of the Enable Multifactor Authentication column for the selected users to false.

These users will no longer need the passcode. However, each user can still enable multifactor authentication for their own user profile. To disable the ability for any user to enable multifactor authentication, disable the system property, disable the system property.

LDAP integration

An LDAP integration allows your instance to use your existing LDAP server as the master source of user data.

Administrators integrate with a Lightweight Directory Access Protocol (LDAP) directory to streamline the user login process and to automate administrative tasks such as creating users and assigning them roles. An LDAP integration allows the system to use your existing LDAP server as the master source of user data. Typically, an LDAP integration is also part of a single sign-on implementation.

The integration uses the LDAP service account credentials to retrieve the user distinguished name (DN) from the LDAP server. Given the DN value for the user, the integration then rebinds with LDAP with the user’s DN and password. The password that the user enters is contained entirely in the HTTPS session. The integration never stores LDAP passwords.

The integration uses a read-only connection that never writes to the LDAP directory. The integration only queries for information, and then updates its internal database accordingly.
Aspects of LDAP integration

Administrators integrate with a Lightweight Directory Access Protocol (LDAP) directory to streamline the user login process and automate administrative tasks such as user creation and role assignment.

User data is refreshed from your master source into the instance. The instance integrates with your organization's internal directory services through an LDAP query as a read-only connection, never updating your corporate LDAP.
Data population

An integration to your LDAP server(s) allows you to quickly and easily populate the instance with user records from your existing LDAP database. In case of data inconsistencies, configuration settings provide the ability to create, ignore, or skip records.

You can also specify the data that is imported by specifying attributes. We recommend importing only the data that you want to expose to the instance.

**Note:** You must specify all the attributes used in your transform map. If attributes are not specified, all available object attributes are imported from the LDAP server. This peripheral data accumulates in temporary import set tables, slowing import time.

For more information, see *Specify LDAP attributes* on page 1832 for configuration information and *Create a transform map* on page 1556 to help you create a transform map.

Authentication

When one of your users enters their domain credentials in the login page, the application passes those credentials to the defined LDAP server(s). The LDAP server responds with an authorized or unauthorized message which the application uses to determine if access should be granted. By authenticating against your LDAP server, users use the same credentials for the application that they use for other internal resources on your domain. Also, you can leverage any existing password and security policies that are already in place (for example: account lockout after a number of failed logins and password expiration dates). Because the application is receiving a “yes” or “no” from the LDAP server, these policies are enforced.

Features of LDAP integration

LDAP integration features include scheduled refresh, a dedicated listener, and on-demand login.

Scheduled LDAP refresh

A scheduled scan of your LDAP server is usually run once a night. It queries all applicable user records’ attributes and compares them with the account on our servers. If there is a difference, we modify our user record with the changed attribute. The load placed upon the LDAP server during the refresh depends on how many records are queried, and the number of attributes being compared. We recommend scheduling the refresh during off-peak hours. A large refresh operation can affect other scheduled operations, such as running reports, and should be planned to minimize any conflicts.

LDAP listener

LDAP listener is our version of a persistent query (or persistent search). We issue a standing query for changes made to your LDAP server, and constantly listen for a response. Assuming your server supports a persistent search, any changes made to any of your applicable LDAP accounts are returned to the LDAP listener and sent to your instance within approximately 10 seconds. This is an extremely useful tool, allowing us to have a nearly real-time copy of your users’ account details, without having to wait for the next scheduled refresh.
On-demand LDAP login

After LDAP integration is complete, your instance has the ability to allow new users to login to the system, even if their accounts have not yet been created. When a new user attempts to login to your instance, we look to see if this user has an account. When the account is not found, the instance automatically queries the LDAP server for the username that was entered. If an account is found, we then try to authenticate with the user's password. If the password checks out, the instance creates an account for the user, populates the account with all applicable LDAP information, and logs the user into your instance.

LDAP data population

An LDAP integration involves data population and authentication.

Note: Functionality described in this integration is not available by default. This integration involves post-deployment customization performed by an experienced administrator or by ServiceNow professional services consultants.

An integration to the LDAP servers allows you to quickly and easily populate the instance's database with user records from the existing LDAP database. To prevent data inconsistencies, configuration settings provide the ability to create, ignore, or skip incoming LDAP records.

You can also limit the data the integration imports by specifying LDAP attributes, thereby importing only the data that you want to expose to an instance. Typically, the LDAP attributes you specify become part of the integration transform map. If you do not specify any LDAP attributes, the integration imports all available object attributes from the LDAP server. The instance stores imported LDAP data in temporary import set tables, so the more attributes you import, the longer the import time. For more information, see Specify LDAP attributes on page 1832.

LDAP scheduled refresh

It is recommended that you run a scheduled scan of the LDAP server once a night.

The scan queries all applicable user records' attributes and compares them to accounts on your instances. If the scan identifies a difference, the integration modifies the instance user record with the changed attribute.

The load placed on the LDAP server during the refresh depends on how many records are queried and the number of attributes being compared.

Schedule the refresh during off-peak hours at a time that minimizes conflicts. A large refresh operation can affect other scheduled operations, such as running reports.

LDAP authentication

Use LDAP authentication to access using LDAP credentials.

When a user enters network credentials in the login page:

1. The instance passes the credentials to an LDAP server to find the instance.
2. With RDNs, it validates the user's DN string. It validates only if at least one of the LDAP OU configurations with table=sys_user has an RDN configured.
3. The LDAP server responds with an authorized or unauthorized message that the system uses to determine whether access should be granted.

By authenticating against your LDAP server, users access the platform with the same credentials that they use for other internal resources on your network domain. Also, you can reuse any existing password and security policies that are already in place. For example, the LDAP server may already have account lockout and password expiration policies.

When you enable LDAP, the system updates user records with these fields.
Table 423: LDAP user record updates

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Identifies whether or not LDAP is used to validate a user. If the source starts with ldap, then the user is validated via LDAP. If the source does not start with ldap, then the password on the user record is used to validate the user upon login.</td>
</tr>
<tr>
<td>LDAP Server</td>
<td>Identifies which LDAP server authenticates the user when there are multiple LDAP servers.</td>
</tr>
</tbody>
</table>

**Note:** The system does not support LDAP password authentication through a MID Server. An instance must be able to directly connect with an LDAP server to support password authentication.

LDAP on-demand login
Create new accounts by matching users to existing LDAP accounts.

After an LDAP integration is established, the instance can allow new users to log in to the system even if they do not yet have an account on the instance. When a new user attempts to log in to the instance, the integration checks to see if this user has an account in the instance. If the integration does not find an existing user account, it automatically queries the LDAP server for the username that was entered. If a matching LDAP account is found, the integration tries to authenticate with the password the user entered. If the password is valid, the instance creates an account for the user, populates the account with all applicable LDAP information, and logs the user in to the instance.

On-demand login uses the LDAP User Import transform map. For more information on transform map requirements, see *LDAP transform maps* on page 1842.

LDAP integration requirements
Review the requirements for LDAP integration, which include a PKI certificate and LDAP compliment directory services server.

LDAP integration requires:
- An LDAP v3 compliant directory services server
  - Allows inbound network access through the firewall (to the LDAP server)
  - *(Optional)* Accepts anonymous login
  - *(Optional)* Supports paging for large LDAP queries
- The external IP address or fully-qualified domain name of the LDAP server. You can also use a MID server.
- A read-only LDAP account of your choosing
- For multiple domains, network access for each domain controller
- For LDAPS, a PKI certificate
- For LDAP listener, a Microsoft Active Directory server that supports persistent queries (ADNotify)

Supported LDAP servers
The instance supports several LDAP servers.

Using JNDI to interface with the LDAP server, the instance has successfully integrated with the following servers:
- Microsoft Active Directory

© 2017 ServiceNow. All rights reserved.
• Novell
• Domino (Lotus Notes)
• Open LDAP

LDAP query limits
There are several methods of handling LDAP limitations.

By default, Active Directory 2000/2003 has an LDAP query limit (maxPageSize) of 1000 objects to prevent excessive loads and denial of service attacks. The system has two methods of dealing with this limit.

• The default method is to break up the query to return fewer than 1000 objects at a time. For example, query only for objects starting with the letter a, then query for b objects.
• The more efficient method for large environments is to enable paging, which is supported by default on all Microsoft Active Directory servers. Paging automatically splits the results into multiple result sets so the integration does not have to split up the query into multiple requests.

LDAP configuration options
There are several optional add-ons you can consider when you configure LDAP.

Secure LDAP connections
Secure connections provide additional protection for an LDAP integration.

The LDAP integration ensures security by connecting from a single machine that uses a fixed IP address through a specific port on the firewall. Furthermore, the connection requires a read-only LDAP account of your choosing for authentication. If you need additional protection for the LDAP integration, you can use one of these security features:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID Server</td>
<td>To shield your LDAP server from external network traffic, install a MID Server on the local network and configure the system to communicate with the MID Server over a secure channel.</td>
</tr>
<tr>
<td>LDAPS</td>
<td>To establish an encrypted LDAPS connection, load the public side of your LDAP server’s SSL certificate. The integration uses the certificate to encrypt all communication between the LDAP server and the instance.</td>
</tr>
<tr>
<td>VPN</td>
<td>To secure the LDAP server with an encrypted point-to-point IPSEC VPN tunnel, speak to your account manager for details and pricing.</td>
</tr>
</tbody>
</table>

For more information about VPNs, Mid Servers, and LDAP integrations, see *You Don’t Need A VPN Part I* on the ServiceNow Community.

LDAP listener
A listener is a dedicated process that periodically searches for changes to users and groups on the LDAP server.

The listener can be deployed on a Microsoft Active Directory server that supports persistent queries (ADNotify), or on an LDAP server that supports persistent search request control (with OID 2.16.840.1.113730.3.4.3)
If the LDAP server supports a persistent search, the LDAP listener recognizes any user and group changes made to any of the applicable LDAP accounts and forwards them to your instance within approximately 10 seconds. This allows the instance to have a nearly real-time copy of your users’ account details without having to wait for the next scheduled refresh. The LDAP listener can only synchronize objects that map to the User [sys_users] and Group [sys_user_group] tables.

**Note:** If a user is added via the listener, but the user does not meet the requirements as defined by the OU filter, then the instance ignores the record on the LDAP server. If it meets the criteria, the user is added to the instance.

**LDAP monitor**
The LDAP monitor provides the current status of the LDAP servers and listener (starting with the Eureka release).

Monitoring the current health of your LDAP configuration can help with LDAP diagnostics and maintenance.

![LDAP Monitor](image)

**Figure 515: LDAP monitor**

The available states are:

- **Active**
• Inactive
• Error
• Active (Shutting down...)
• Error (Shutting down...)

In addition to its current state, the monitor also shows:
• The last message detected by the listener, such as waiting for LDAP changes, error connecting, and so forth.
• The last LDAP user change, such as new user, updated user, and so forth.
• The last error that occurred.

View LDAP monitor
You can view current information about LDAP servers and listeners using LDAP monitor.

Role required: admin

To view LDAP monitor:

Navigate to **LDAP > System LDAP > LDAP Monitor.**

See the table for descriptions of the properties and fields in the screen.
Table 425: LDAP monitor

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh</td>
<td>You can configure the refresh rate by clicking the <strong>Refresh</strong> field in the LDAP Server Monitor header bar, and selecting the number of seconds between each data refresh. You can also select <strong>None</strong> to suppress refreshing.</td>
</tr>
<tr>
<td>Connection Status</td>
<td>The server connection indicator is located on the right side, above the LDAP Listener Status fields. When the server is connected, the box is green and shows Connected. When the server is not connected, the box is red and shows Not Connected. When the server connection is being tested, the box is yellow and shows Testing Connection.</td>
</tr>
<tr>
<td>LDAP Server Properties</td>
<td>As you monitor LDAP servers, you can make changes to the properties by clicking <strong>Edit</strong> in the LDAP Server Monitor screen.</td>
</tr>
</tbody>
</table>
| Server URL       | The combination of the server name and server port where the LDAP Server is listening. Frequently, the port is set to one of the following:  
  • 389: the default port for connecting to LDAP in clear text  
  • 636: the standard port for connecting to LDAP via an SSL connection  
  Example value: ldap://10.10.10.3:389/  
  Your LDAP Server may have more than one URL address. This does NOT establish multiple directory structures from which you can import data, which is done by creating another LDAP Server entry, but does provide for redundancy when you have multiple LDAP Servers to avoid a single point of failure. The LDAP URL addresses are separated with a space character, and the system automatically tries each server address in turn until a valid connection can be made. |
Field | Description
--- | ---
Starting search directory | The starting directory or RDN (Relative Distinguished Name) where the system begins searching for users or groups. Example value: DC=service-now,DC=com

No data ABOVE this point is available for import. The instance has visibility into the specified directory and directories BELOW it in the LDAP hierarchy.

MID Server Status | The current connection status of the MID Server.

LDAP Listener Status

Current Status | This indicates whether the listener is active.

Last Info Message | This shows the last message the LDAP server received relating to user and group changes, and the time the message was received.

Last Change | This shows the last change made to the LDAP server, and the time it was made.

Last Error | This shows the last error that occurred on to the LDAP server, and the time it occurred.

**Multiple LDAP domains**

You can establish multiple network domains within the same forest or for completely non-trusted domains.

The recommended method for handling multiple domains is to create a separate LDAP server record for each domain. Each LDAP server record must point to a domain controller for that domain. This means the local network must allow connections to each of the domain controllers.

After expanding to more than one network domain, it is critical that you identify unique LDAP attributes for the application user names and import coalesce values. A common unique coalesce attribute for Active Directory is `objectSid`. Unique user names may vary based on the LDAP data design. Common attributes are `email` or `userPrincipalName`.

**LDAP integration FAQs**

Review these frequently-asked questions about LDAP integration in the instance.

**What are the prerequisites for an LDAP integration?**

- The directory services server must be LDAP v3 compliant
- Inbound network access through the firewall must be allowed (to the LDAP server)
- External IP or Name of the LDAP server
- User credentials with read-only access
- For LDAPS, a PKI certificate
When is an LDAP integration usually done?

LDAP integrations are usually done before the instance Go Live, but can be integrated at any time.

Is this a synchronization or a copy?

This question comes up regularly during our pre-integration discussions, and is centered around a concern of a third party (the instance in this case) making changes (writing) to your LDAP server. In an LDAP integration, your instance does not write to the internal LDAP directory. The instance queries for information, and updates its database accordingly.

No changes are made to the internal LDAP server by the instance. The service account is read only.

Is it secure?

Yes. The connection is made from a single machine using a fixed IP address through a specific port on your firewall. Authentication is done with a read-only LDAP account of your choosing. You can use standard LDAP, or load the public side of an SSL certificate installed on your directory, in which case we can use LDAPS. To add another layer of security, we also offer the option of a point-to-point IPSEC VPN tunnel. Speak to your account manager for details and pricing.

Another security aspect to consider is the data shared in an LDAP integration. To limit the data exposed to your instance, specify attributes in your transform map. For more information, see Create a transform map on page 1556.

How up to date is the information?

Most changes (including additions) to your LDAP server are available to the instance within seconds, depending on how many components of the full LDAP integration are in place.

Which attributes need to be pulled from the directory into the instance?

It is recommended that attributes are defined to import only required data. Defined attributes get mapped into the instance user database.

We cannot answer the question of which specific attributes are needed because this is determined by the scope of the project and business requirements.

What types of LDAP servers does the instance support?

The instance has successfully integrated with Microsoft Active Directory, Novell, Domino (Lotus Notes), and Open LDAP. We use JNDI to interface with the LDAP Server. As long as your LDAP server is LDAP v3 compliant, the integration is successful.

Since my users are already authenticated on my local network, how can I keep them from having to enter a password to access the application?

A single sign-on method is the solution. Along with the data population functionality provided with the LDAP import, you can use the External Authentication functionality supported by the application.
Can I integrate with multiple domains?

Yes, multiple domains can be within the same forest or completely non-trusted domains. The recommended method is to create a separate LDAP server record for each domain. Each LDAP server record must point to a domain controller for that given domain. This means that connections must be allowed to each of the domain controllers.

When you expand to more than one domain, it is critical that you identify unique LDAP attributes to be used as the application user names and import coalesce values. A common unique coalesce attribute for Active Directory is objectSid. Unique user names may vary based on your LDAP data design; common attributes are email or userPrincipalName.

How do you handle querying more than 1000 users?

By default, Active Directory 2000/2003 has an LDAP query limit (maxPageSize) of 1000 objects to prevent excessive loads and denial of service attacks. We have two methods of dealing with this limit.

The default method is to break up the query to return less than 1000 objects at a time. For example, query only for object starting with the letter 'a', then query for 'b' objects. The more efficient method for large environments is to enable paging. Paging is supported by default on all Microsoft Active Directory servers. It automatically splits the results into multiple result sets, so we don't have to split up the query into multiple requests.

What type of LDAP query is done?

If an LDAP password is supplied then a "Simple Bind" is performed. If no LDAP password is supplied then "none" is used, in which case the LDAP server must allow anonymous login.

How is LDAP authentication accomplished when the username is provided?

We use provided service account credentials for LDAP to retrieve the user DN from the LDAP server. Given the DN value for the user, we then rebind with LDAP given the user's DN and the provided password.

How is the user password stored?

The password that the user enters is contained entirely in their HTTPS session. We do not store that password anywhere.

Are LDAP records synchronized or just copied?

The instance does not synchronize department records. Users and group memberships are kept up-to-date by the LDAP Listener mechanism and a daily full LDAP Browse, but the instance does not delete any of these entries once they disappear from LDAP.

If an entry were to be deleted, the entire history would also get deleted, and any references to it would be cleared or deleted. Configuration Items (CIs), SLA Agreements, Software Licenses, Purchase Orders, and Service Catalog Entries all have a reference to Department, and if Department is deleted, then those references get cleared. There are many references to Users, and so deleting a user would lose all history of what that user did. Currently, the decision to delete or not to delete is made by our customers.
How is a user record defined to use LDAP authentication?

These fields on the user record pertain to LDAP:

- **Source:** The Source field identifies whether or not a user is validated using LDAP. If the source field starts with "ldap", then the user is validated via LDAP. If the Source field does not start with "ldap", then the password on the user record is used to validate the user upon login.

- **LDAP Server:** The instance supports multiple LDAP servers, so the LDAP Server field determines which server should be used to authenticate the user.

How can we keep LDAP records synchronized?

Schedule a periodic scan of the LDAP server to pick up changes.

I'm ready to configure my LDAP integration. Now what?

Let's go! Start with *LDAP integration setup* on page 1821.

**LDAP integration setup**

Administrators can enable LDAP integration to allow single sign-on of users from their company LDAP directory.

After the integration, the MID Server connects to the instance and the MID Server also connects to the LDAP server. In both cases, the MID Server initiates the connection:

1. First, the MID Server connects to the LDAP server via LDAP on Port 389.

2. Then, the MID Server initiates an HTTPS encrypted connection to the instance on Port 443 to push the data to the instance.

*LDAP communication channels*

LDAP typically uses one of these types of communication channels.
<table>
<thead>
<tr>
<th>Connection</th>
<th>Description</th>
<th>LDAP import support?</th>
<th>LDAP authentication support?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID Server connection</td>
<td>Communicates over HTTP on port 80 by default. This communication channel does not require a certificate. The connection between the MID Server and the instance is over HTTPS (port 443). You can use the MID Server to import data over LDAP, but you cannot use the MID Server for LDAP authentication. Proceed to <em>Define the LDAP Server.</em></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Standard LDAP integration</td>
<td>Communicates over TCP on port 389 by default. This communication channel does not require a certificate. Proceed to <em>Define the LDAP Server.</em></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SSL-encrypted LDAP integration (LDAPS)</td>
<td>Communicates over TCP on port 636 by default, This communication channel requires a certificate. Proceed to <em>Upload the LDAP X.509 SSL certificate</em> on page 1823 to obtain and upload the certificate.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VPN connection</td>
<td>Communicates over an IPSEC tunnel. Purchase or create an IPSEC tunnel on your local network. Proceed to <em>Define the LDAP Server.</em></td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

A MID server initiates one connection to an LDAP server via port 389, then initiates an encrypted HTTPS connection to an instance via port 443 to push data to the instance. When using a MID server, the instance does not make the connection to the LDAP server. The MID server does.

The instance can also connect to the LDAP server directly, using LDAP or LDAPS, either over the internet or through a VPN tunnel.
For more information about VPNs, Mid Servers, and LDAP, see *You Don't Need A VPN Part II* on the community.

**Upload the LDAP X.509 SSL certificate**
If your administrator is setting up an SSL-encrypted LDAP integration (LDAPS) to communicate over TCP on port 636, and has not already uploaded a certificate as part of your instance Go Live activities.

Role required: admin

1. Purchase or generate an SSL certificate on your LDAP server.
2. Upload the LDAP certificate to the instance.

**Define an LDAP server**
Create a new LDAP server record in the instance.

Role required: admin

1. Navigate to **System LDAP > Create New Server**.
2. Fill in the form fields.
3. Click Submit.

Note: You can also modify an existing LDAP server record by navigating to System LDAP > LDAP Servers and making the needed changes.

LDAP server fields
You can configure LDAP servers after you create a new server record.
Figure 516: LDAP server form

Table 427: LDAP connection properties

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the server.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box if the server is active.</td>
</tr>
<tr>
<td>LDAP Server URLs</td>
<td>Enter the URLs of the primary and backup LDAP servers. Servers are first ordered by operational status, with servers that are Up listed first, then ordered by the Order value that you specify. The first server listed is the primary LDAP server. The others are redundant servers.</td>
</tr>
<tr>
<td>Server URL</td>
<td>Enter the URL of the server. Configure the form to add this field if necessary. It is a calculated read-only field that shows the list of LDAP servers that you can also see in the <strong>LDAP Server URLs</strong> field, separated by a space, and ordered by operational status and the order values of the URLs.</td>
</tr>
<tr>
<td>Login distinguished name</td>
<td>Enter the distinguished name (DN) of the user authenticating the LDAP connection.</td>
</tr>
<tr>
<td>Login password</td>
<td>Enter the server's password.</td>
</tr>
<tr>
<td>Starting search directory</td>
<td>Enter the relative distinguished name (RDN) of the default search directory. All queries to this LDAP server will start from this RDN.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| MID Server   | Select the MID Server you want to use to connect to the LDAP server. Using a MID Server to establish an LDAP connection prevents you from having to expose the LDAP server to external network traffic. It also eliminates the need to establish a VPN tunnel between your LDAP server and ServiceNow data centers.  
  **Note:**  
  - The MID Server user must have the user_admin role in order to be able to read LDAP server configuration records.  
  - The following are not available with the MID Server:  
    - LDAP authentication  
    - SSL connection |
<p>| Connect timeout | Specify the maximum number of seconds that the instance has to establish an LDAP connection. If no connection is made by this time, the connection is terminated. |
| Read timeout  | Specify the number of seconds the integration has to read LDAP data. The integration stops reading LDAP data after the connection exceeds the read timeout. If you enable an SSL connection, you can also set a read timeout value with the <code>com.glide.ssl.read.timeout</code> system property. If you enter timeout values for both this field and the system property, the lowest timeout value takes precedence. |
| SSL           | Select this check box to require the LDAP server to make an SSL-encrypted connection. For more information, see Enable SSL. If you selected a MID Server, this field is not available. |
| Listener      | Select this check box to enable the integration to periodically poll Microsoft Active Directory servers or LDAP servers that support persistent search request control. Additionally, if you selected a MID Server, the listener functionality is available for that MID Server. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listener</td>
<td>Select this check box to enable the integration to periodically poll Microsoft Active Directory servers or LDAP servers that support persistent search request control. Additionally, if you selected a MID Server, the listener functionality is available for that MID Server. See <a href="#">LDAP listener</a> on page 1814 and <a href="#">Enable an LDAP listener</a> on page 1831 for more information.</td>
</tr>
<tr>
<td>Listen interval (timeout value)</td>
<td>Specify the listener timeout value in the number of minutes that the integration listens for LDAP data with every connection. The integration stops listening for LDAP data after the connection exceeds the listen interval.</td>
</tr>
<tr>
<td>Paging</td>
<td>Select this check box to have the LDAP server split up LDAP attribute data into multiple result sets rather than submit multiple queries.</td>
</tr>
</tbody>
</table>

Specify a redundant LDAP server
The LDAP integration uses a redundant server if the primary LDAP server experiences a service interruption.

Role required: admin

Administrators can specify redundant servers from either the Create New Server module or from an individual LDAP Server record. The instance searches for an available LDAP server in the order in which they are listed.

1. Navigate to **System LDAP > Create New Server**.
2. Fill out the form as specified in *Define the LDAP Server*.
3. In the **Server URL** field, the valid URLs of all servers appear separated by a space. Servers are first ordered by operational status, with servers that are **Up** listed first, then ordered by the **Order** value that you specify. The first server listed is the primary LDAP server. The others are redundant servers.

    **Note:** There is a slight delay between the change in the actual operational status and the display.

4. Enter other LDAP server fields as needed.
5. Click **Submit**.

© 2017 ServiceNow. All rights reserved. 1827
Specify a redundant LDAP server from an LDAP server record
The LDAP integration uses a redundant server if the primary LDAP server experiences a service interruption.

Role required: admin

1. Navigate to System LDAP > LDAP Servers.
2. Select the LDAP server for which you want to specify a redundant server.
3. From the LDAP Server URLs embedded list, click Insert a new row.
4. Fill in the fields for the row (see table).
5. Right-click the form header and click Save.
6. Repeat these steps for each additional server you want to specify.
Set up SSL-encrypted LDAP integration (LDAPS)
Set up an SSL connection for your LDAP server.
Role required: admin
If you use an LDAPS integration and the default SSL port is 636, no further configuration is necessary; SSL is automatically enabled. If the LDAPS integration uses another SSL port, define the alternate SSL connection properties.

1. Navigate to **System LDAP > LDAP Servers**
2. Select the LDAP server to configure.
3. Under **Related Links**, click **Advanced view**.
4. In the **Server URL** field, specify the LDAP IP address and alternate SSL communications port.
5. Select the **SSL** check box.
   If this option does not appear on the form by default, configure the form and add it.
6. Click **Update**.

**Note:**
Be sure a network administrator configures the local firewall to allow the application server to access the LDAP server. If the LDAP server is located within an internal network, the firewall forwards (or NATs) the application server's IP address through the firewall on the correct port.

---

Provide LDAP server login credentials
The LDAP login credentials determine what organizational units the integration can see.
Role required: admin
Servers that allow anonymous login generally limit the organizational unit (OU) data available to anonymous connections.

1. Navigate to **System LDAP > LDAP Servers**.
2. Select the LDAP server to configure.

---

**Table 428: LDAP Server URLs embedded list**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>The URL or IP address to the redundant LDAP server.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the instance searches for an available LDAP server from</td>
</tr>
<tr>
<td></td>
<td>lowest value to highest. A business rule automatically populates this value</td>
</tr>
<tr>
<td></td>
<td>if you leave the field blank.</td>
</tr>
<tr>
<td>Active</td>
<td>A true/false field indicating whether the LDAP server is available for use</td>
</tr>
<tr>
<td></td>
<td>as a backup server. Only active servers can be used as backup servers.</td>
</tr>
<tr>
<td>Operational Status</td>
<td>A read-only true/false field indicating whether the LDAP server is currently available. Only servers that are currently operational can be used as backup servers.</td>
</tr>
</tbody>
</table>
3. In Login distinguished name, enter the user credentials for an account with read access to the directory levels from which you want to import users or groups. The system uses these credentials to connect to your LDAP server. If this information is not entered, the application attempts an anonymous login to the LDAP server.

The Login distinguished name fields accepts several formats.
To access a Microsoft Active Directory (AD) server, use one of the following:

- user@domain.com, domain\user
- cn=user,ou=users,dc=domain,dc=com>

To access a different LDAP directory server, the username must be in the full distinguished name format:

- cn=user,ou=users,dc=domain,dc=com

4. In Login password, enter the password for the LDAP user.

   **Note:** Consider enabling LDAPS to encrypt this password during transmission.

5. Select the Active check box.

6. [Optional] In the Starting search directory field, explicitly specify the LDAP OU attributes you want the instance to import.

7. Click Update.

   **Note:**
   If you provide an LDAP password, the integration performs a Simple Bind operation. If you do not provide an LDAP password, the LDAP server must allow anonymous login or the integration cannot bind to the LDAP server.

---

Enable an LDAP listener

Enabling a listener is optional. If enabled, a listener notifies the system to process LDAP records soon after there is an update on the LDAP server.

Role required: admin

A listener is a dedicated process that periodically searches for changes on the LDAP server.

The listener can be deployed on a Microsoft Active Directory server that supports persistent queries (ADNotify), or on an LDAP server that supports persistent search request control (with OID 2.16.840.1.113730.3.4.3).

To enable a listener:

1. Navigate to System LDAP > LDAP Servers.
2. Select the LDAP server to configure.
3. Select the Listener check box.
4. Click Update.

**Note:**

The system only imports user records that match the LDAP OU filter. Incoming user records that do not meet the filter requirements are flagged as invalid and ignored by the import. Administrators can enable verbose LDAP logging to determine if incoming records are not matching the LDAP OU filter.
Specify LDAP attributes

By configuring the LDAP Server form and adding the **Attributes** field, you can specify, and thereby limit, the attributes the LDAP server query returns. This can enhance performance as well as security.

Role required: admin

By default, the system loads all of the attributes for each object that it has permission to read from your LDAP server. By configuring the LDAP Server form and adding the **Attributes** field, you can specify, and thereby limit, the attributes the LDAP server query returns. Using this approach for large LDAP imports can greatly improve the speed of those imports.

For best results, define attributes where possible. If there is information that you do not want exposed to the system, exclude the attribute.

If you do not specify LDAP server attributes, user transactions may freeze for extended periods of time when new attributes are added to an LDAP server object because the system will be busy loading data from the new attributes.

**Note:** To use the manager lookup scripts described in Select or Create a Transform Map for LDAP Data, specify **manager** and **dn** (distinguished name) in the **Attributes** field. Neither attribute is required to be a part of a transform map.
Set LDAP connection properties
Configure your LDAP server connection properties.

Role required: admin

1. Navigate to System LDAP > LDAP Servers.
2. Select the LDAP server to configure.
3. Set the connection property fields (see table).
4. Click Update.

Table 429: LDAP connection properties

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the server.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box if the server is active.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LDAP Server URLs</td>
<td>Enter the URLs of the primary and backup LDAP servers. Servers are first ordered by operational status, with servers that are Up listed first, then ordered by the Order value that you specify. The first server listed is the primary LDAP server. The others are redundant servers.</td>
</tr>
<tr>
<td>Server URL</td>
<td>Enter the URL of the server. Configure the form to add this field if necessary. It is a calculated read-only field that shows the list of LDAP servers that you can also see in the <strong>LDAP Server URLs</strong> field, separated by a space, and ordered by operational status and the order values of the URLs.</td>
</tr>
<tr>
<td>Login distinguished name</td>
<td>Enter the distinguished name (DN) of the user authenticating the LDAP connection.</td>
</tr>
<tr>
<td>Login password</td>
<td>Enter the server's password.</td>
</tr>
<tr>
<td>Starting search directory</td>
<td>Enter the relative distinguished name (RDN) of the default search directory. All queries to this LDAP server will start from this RDN.</td>
</tr>
<tr>
<td>MID Server</td>
<td>Select the MID Server you want to use to connect to the LDAP server. Using a MID Server to establish an LDAP connection prevents you from having to expose the LDAP server to external network traffic. It also eliminates the need to establish a VPN tunnel between your LDAP server and ServiceNow data centers.</td>
</tr>
</tbody>
</table>

**Note:**
- The MID Server user must have the user_admin role in order to be able to read LDAP server configuration records.
- The following are not available with the MID Server:
  - LDAP authentication
  - SSL connection

<p>| Connect timeout       | Specify the maximum number of seconds that the instance has to establish an LDAP connection. If no connection is made by this time, the connection is terminated. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read timeout</td>
<td>Specify the number of seconds the integration has to read LDAP data. The integration stops reading LDAP data after the connection exceeds the read timeout. If you enable an SSL connection, you can also set a read timeout value with the <code>com.glide.ssl.read.timeout</code> system property. If you enter timeout values for both this field and the system property, the lowest timeout value takes precedence.</td>
</tr>
<tr>
<td>SSL</td>
<td>Select this check box to require the LDAP server to make an SSL-encrypted connection. For more information, see Enable SSL. If you selected a MID Server, this field is not available.</td>
</tr>
<tr>
<td>Listener</td>
<td>Select this check box to enable the integration to periodically poll Microsoft Active Directory servers or LDAP servers that support persistent search request control. Additionally, if you selected a MID Server, the listener functionality is available for that MID Server.</td>
</tr>
<tr>
<td>Listen interval (timeout value)</td>
<td>Specify the listener timeout value in the number of minutes that the integration listens for LDAP data with every connection. The integration stops listening for LDAP data after the connection exceeds the listen interval.</td>
</tr>
<tr>
<td>Paging</td>
<td>Select this check box to have the LDAP server split up LDAP attribute data into multiple result sets rather than submit multiple queries.</td>
</tr>
</tbody>
</table>

Automatic LDAP server validations

When an LDAP Server record is set to active, the system automatically tests every connection to validate it.

Validations include:

- The LDAP server is accessible at the provided URL and port
- The LDAP server URL is properly formatted
• The login credentials are valid

Starting with the Fuji release, the system displays colored dots next to each server URL:

**Table 430: LDAP server connection icons**

<table>
<thead>
<tr>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>The server if active and operational.</td>
</tr>
<tr>
<td>Gray</td>
<td>The server is neither active nor operational.</td>
</tr>
<tr>
<td>Red</td>
<td>The server is active but not operational.</td>
</tr>
</tbody>
</table>
Figure 517: LDAP server connection status

Test an LDAP connection manually
You can manually test the connection to the LDAP server from the LDAP server form.
Role required: admin

1. Navigate to System LDAP > LDAP Servers.
2. Select the LDAP server to test.
3. Under Related Links, click Test connection.
4. Under Related Links, click Browse to verify that the appropriate LDAP directory structure is visible to the system.
5. (Optional) If the connection was successful, click Browse to view the source LDAP directory structure that is visible to the instance.

Note:
The Filter and RDN fields on the left of the Browse window are ignored when you use the search field on the right.

Test communication between LDAP and MID Servers
The instance tests the connection automatically every time a user opens the LDAP Server form.

Role required: admin

The instance supports an LDAP connection timeout of 29 seconds or less. Error messages appear on the form if there are any issues connecting to the LDAP server.

Employees can also verify connectivity between the instance and the LDAP server. Contact Technical Support for assistance verifying LDAP connectivity.

LDAP connection timeout
The instance tests the connection automatically every time a user opens the LDAP Server form.

Error messages appear on the form if there are any issues connecting to the LDAP server.

Note: Employees can also verify connectivity between the instance and the LDAP server. Contact Technical Support for assistance verifying LDAP connectivity.

Automatic LDAP operational status update
The instance changes the operational status of LDAP servers depending on the result of the connection test.

- If your instance establishes a connection to a server that has a Operational Status value of down, the Operational Status value is automatically changed to up. This functionality is supported for both automatic and manual connection tests.
- If a connection cannot be established to a server that has a Operational Status value of up, the Operational Status value is automatically changed to down. This functionality is supported for automatic connection tests only, not manual tests.

Define LDAP organizational units
An organizational unit (OU) definition specifies the LDAP source directories available to the integration.

Role required: admin

OU definitions can contain locations, people, or user groups. Every LDAP server definition contains two sample OU definitions: one for importing groups into the system and the other for users.

1. Navigate to System LDAP > LDAP Servers.
2. Select the LDAP server to configure.
3. In the LDAP OU Definitions related list, select either the Groups or Users sample OU definition.
4. Complete the LDAP OU Definition form (see table).
5. Click Update.
The system automatically tests the connection to the LDAP server.

6. Under Related Links, click Browse to view the LDAP directory records that the OU definition returns.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name the integration uses when referencing this OU. The name you enter here becomes an LDAP target in the data source record.</td>
</tr>
<tr>
<td>RDN</td>
<td>Specify the relative distinguished name of the subdirectory you want to search. This RDN is combined with the start-searching directory from the LDAP server definition to identify the subdirectory containing information for this organizational unit. For example, the sample OU definition uses the RDN value of CN=Users to search the LDAP directory CN=Users,DC=service-now,DC=com and any directory below this point. This field must match a subdirectory in your LDAP system.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Query field</td>
<td>Specify the name of the attribute within the LDAP server to query for records. The query field must be unique in both single and multiple domain instances. For best results, use email addresses or other credentials that uniquely identify the user in a multiple domain instance. Active Directory uses the sAMAccountName attribute. Other LDAP servers tend to use the cn attribute.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The Query field must map to the User ID field in the User [sys_user] table. For example, if an Active Directory user logs in as joe.example, there must be a user record with a User ID value of joe.example and an LDAP record with an sAMAccountName value of joe.example.</td>
</tr>
</tbody>
</table>
Further suppose that you want to exclude the HR group and HR users from the application. Do the following:

1. Create an LDAP server record with a starting search directory of `dc=my-domain,dc=com`.
2. Create an OU definition record for `ou=Groups` with a filter to exclude `cn=HR`.
3. Create an OU definition record for `ou=Users` with a filter to exclude `ou=HR`.

If you do not specify additional attributes or filters with an OU definition, the LDAP query returns the entire sub-tree from the starting directory and RDN.

In these examples, an OU definition with the RDN value of `ou=Groups` and no filter would have returned all groups. Likewise, an OU definition with the RDN value of `ou=Users` and no filter would have returned all users and child organizational units.

**Create an LDAP OU data source**

Each LDAP organizational unit (OU) definition has its own related list of data sources.

Role required: admin

---

**Note:** Both the **LDAP Server** and **LDAP OU Definition** must be active for the test load action to function properly. When the test load is activated for the first time, the system samples up to 20 records to determine the length of the import set fields. If the sampled records do not contain values for the **User ID** field, the system sets the field length for all subsequent imports to the default length of 40. The import truncates any imported data that exceeds the import set table field length. Additionally, the **User ID** field is truncated to a maximum of 40 characters. Be aware that the 20 loaded records cannot be transformed and are for testing purposes only. If the test records contain values for the **User ID** field, the field length is set based on the field length of the longest user ID in the test records.

To create a new data source:

1. Navigate to **System LDAP > LDAP Servers**.
2. Select the LDAP server to configure.
3. In the **LDAP OU Definitions** related list, select an item, such as **Groups** or **Users**.
4. In the **Data Sources** related list, click **New**.
5. Complete the Data Source form (see table).
6. Click **Submit**.
7. Under **Related Links**, click **Test Load 20 Records** to test whether the data source can bring LDAP data into the import table.
Table 432: Data Sources form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name the integration uses when referencing this data source.</td>
</tr>
<tr>
<td>Import set table name</td>
<td>Enter the name of the staging table where the system temporarily places the imported LDAP records and attributes. Review this table to view imported LDAP records. You can use the same import set table name for all LDAP data sources.</td>
</tr>
<tr>
<td>Type</td>
<td>Select LDAP to indicate the imported data is LDAP data. After you select the type LDAP, the form displays the LDAP target field.</td>
</tr>
<tr>
<td>LDAP target</td>
<td>Select the LDAP OU definition associated with this data source.</td>
</tr>
</tbody>
</table>

**LDAP transform maps**

The transform map moves data from the import set table to the target table (User or Group).

The LDAP integration uses standard import sets and transform maps. You can also create custom LDAP transform maps.

**Important**: Whether you select or create custom LDAP transform maps, there should be one active transform map for a set of source and target tables. Enabling multiple transform maps for the same source and target tables can produce duplicate entries in the target table unless you coalesce against the matching fields.

**Default LDAP transform maps**

By default, the system provides two transform maps for LDAP data.

Table 433: Default LDAP transform maps

<table>
<thead>
<tr>
<th>Transform Map</th>
<th>Source Table</th>
<th>Target Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP User Import</td>
<td>ldap_import</td>
<td>sys_user</td>
<td>Default transform map for creating user records from LDAP credentials as part of LDAP on-demand login. Contains mappings for an Active Directory LDAP server.</td>
</tr>
</tbody>
</table>
**LDAP Group Import**

<table>
<thead>
<tr>
<th>Transform Map</th>
<th>Source Table</th>
<th>Target Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Group Import</td>
<td>ldap_group_import</td>
<td>sys_user_group</td>
<td>Default transform map for creating group records from LDAP OUs. Contains mappings for an Active Directory LDAP server.</td>
</tr>
</tbody>
</table>

**Note:** By default, the system does not have a transform map for LDAP department records.

**Requirements for custom LDAP transform maps**

If you choose to create a custom transform map, the transform map must meet the following mapping requirements.

**Table 434: Requirements for custom LDAP transform maps**

<table>
<thead>
<tr>
<th>Source Table</th>
<th>Source Field</th>
<th>Target Table</th>
<th>Target Field</th>
<th>Coalesce</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldap_import</td>
<td>u_source</td>
<td>sys_user</td>
<td>source</td>
<td>false</td>
<td>The <strong>u_source</strong> field identifies the LDAP DN of the imported user or group. The system uses this field to determine that a user requires LDAP authentication, to find a user's manager, and to put users into groups.</td>
</tr>
</tbody>
</table>
| ldap_import    | Select one of the following fields:  
  • u_samaccountname  
  • u_dn  
  • u_cn | sys_user     | user_name    | true     | If LDAP integrates to Active Directory, select **u_samaccountname** as the source field. If other LDAP directories are used, select **u_dn** or **u_cn** as the source field. |
**LDAP data transformation**

If an LDAP attribute contains simple data, the transform map links an imported LDAP attribute to an appropriate field in the target table (User or Group).

For example, sample data in the sAMAccountName attribute maps to the User ID field in the User table.

If the imported LDAP data maps to a reference field, the instance searches for an existing matching record. If no matching record exists, the instance creates a new record for the reference field unless the field mapping specifies otherwise.

For example, suppose the LDAP attribute l maps to the Location reference field in the User table. Whenever the import brings in an attribute value that does not match an existing location record value, the transform map creates a new location record. The new location record has the same value as the imported attribute, and the imported user record now has a link to the new location record.

However, there are times when LDAP attribute returns a distinguished name (DN), which is essentially a reference to another record within the LDAP directory. For example, the manager attribute typically contains the distinguished name for the manager of the current LDAP directory entry. An imported DN typically uses a long text string such as: cn=Beth Anglin,ou=Users,dc=my-domain,dc=com.

**Warning**: Make sure your target fields are long enough to contain a DN. Many text fields use the default length of 40, which may not be long enough for some DN values. The ServiceNow system truncates any value that exceeds the field length.

Administrators do not typically want the system to create new users from the DN value because the new user has no association with an existing user. Instead, administrators want the import to locate the manager’s existing user record and associate it with the newly imported user. The LDAPUtils script include contains the setManager and processManagers functions that can parse a DN and search for an existing user. For best results, use these functions to create a custom transform map.

For example, the LDAP User Import transform map script calls the setManager function:

```java
// The manager coming in from LDAP is the DN value for the manager.
// The line of code below will locate the manager that matches the DN value and set it into the target record. If you are not interested in getting the manager from LDAP then remove or comment out the line below
ldapUtils.setManager(source , target );
```

In some cases, the integration imports a user’s record before importing the associated manager’s user record. To handle such cases, you may want to call the processManagers function after the transform completes. For example, the LDAP User Import transform map uses an onComplete transform script to call the processManagers function.

```java
// It is possible that the manager for a user did not exist in the database when the user was processed and therefore we could not locate and set the manager field. // The processManagers call below will find all those records for which a manager could not be found and attempt to locate the manager again. This happens at the end of the import and therefore all users should have been created and we should be able to locate the manager at this point
ldapUtils.processManagers();
```

Remove or comment out the setManager and processManagers function calls if your LDAP integration does not use the manager attribute.

**onStart and onAfter LDAP scripts**

Any custom transform map should include onStart and onAfter scripts.
The `onStart` script should call the `LDAPUtils` script include and start logging. For example, the LDAP User Import transform map has an `onStart` script that uses this code:

```javascript
gs.include ( "LDAPUtils" ) ; var ldapUtils  = new LDAPUtils ( ) ;
ldapUtils.setLog (log ) ;
```

The `onAfter` script should call the `addMembers` function. For example:

```javascript
ldapUtils.addMembers (source , target ) ;
```

### Scheduled data imports for LDAP
A scheduled import allows administrators to import LDAP data on a regular schedule.

By default, the LDAP integration includes two sample scheduled imports:

- Example LDAP User Import
- Example LDAP Group Import

Neither example is active by default. Change these scheduled imports to meet your company's business needs.

### Auto provision LDAP users
You automatically provision users who are in the LDAP server but not yet in your instance.

#### Role required: admin
- Create the following properties in the System Properties `[sys_properties]` table:

<table>
<thead>
<tr>
<th>LDAP property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>glide.ldap.authentication</code></td>
<td>Enables LDAP authentication by using LDAP to authenticate users. Set this property to <code>true</code> (the default value).</td>
</tr>
<tr>
<td><code>glide.ldap.user.autoprovison</code></td>
<td>Enables LDAP the system to automatically create users in the User <code>[sys_user]</code> table when the user exists in LDAP but is not yet in the instance. Set this property to <code>true</code> (the default value).</td>
</tr>
</tbody>
</table>

Both of these properties must be set to `true` for auto-provisioning to work.

### LDAP import maps
LDAP import maps match fields in your LDAP database to fields in your instance.

**Note:** LDAP mapping has a performance effect, so the recommended approach is to schedule it during off-peak hours, or process a few records at a time to maintain system availability.

The best practice is to define a transform map that only imports the needed or required attributes. Depending on the version of the instance you are using, the method for specifying LDAP mapping relationships varies.

The easiest way to know whether or not you are running a version which uses the System LDAP application for LDAP integration is to find the application from the application navigator.
The **Run Business Rules** option is applied only for the target table. Only transform maps associated to the target table run the business rules associated with different tables. If you are updating a user group and have business rules running on a user group table, the group must have roles define.

### Table 436: LDAP import mapping options

<table>
<thead>
<tr>
<th>System LDAP application?</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Use a <em>transform map</em> to specify your mapping.</td>
</tr>
<tr>
<td>No</td>
<td>Use a LDAP <em>legacy import map</em> to specify your mapping, or the default LDAP transform that is included in baseline instances. Remember to adjust the <strong>Coalesce</strong> field to match against the correct fields.</td>
</tr>
</tbody>
</table>

* Differences between LDAP transform maps and legacy import maps *

When specifying LDAP mapping relationships using transform maps, there is a major difference in how reference fields are set for manager and department.

When using a *transform map*, it is necessary to use a *transform script* to create references. This is because the value associated with an LDAP attribute like “manager” is the distinguished name (DN) of the manager.

Without some extra logic in place, the result is the creation of a user record with a manager name that is the distinguished name of that user in LDAP. The integration includes a transform script to facilitate the creation of these references. The default transform map “LDAP User Import” includes transform scripts for these references.

**Existing mapping relationships**

When updating legacy import maps to transform maps, you can retain the LDAP mapping relationships that existed prior to the addition of the System LDAP application. The LDAP server has a **Map** field that is a reference to the legacy import map.

**Note:** By default this field is hidden, so you have to configure the form to display it.

If you want to transition to using a transform map, clear the reference to the legacy import map.

**LDAP import map settings**

Verify and use attributes to limit the fields the integration imports from the LDAP source. Additionally, it is important to map the user_name field to the LDAP attribute that contains the user's login ID. For Active Directory this is usually the sAMAccountName attribute. If you would like to import and coalesce on a binary attribute (such as objectSID or objectGUID), you have to create a custom transform script.

**Note:** Any value mapped to the user_name field must be unique.

If you do not specify a transform map (such as LDAP User Import), the integration uses the following default mappings:
### Table 437: LDAP import default mapping

<table>
<thead>
<tr>
<th>User field or variable</th>
<th>LDAP attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name</td>
<td>sAMAccountName</td>
</tr>
<tr>
<td>email</td>
<td>mail</td>
</tr>
<tr>
<td>phone</td>
<td>telephoneNumber</td>
</tr>
<tr>
<td>home_phone</td>
<td>homePhone</td>
</tr>
<tr>
<td>mobile_phone</td>
<td>mobile</td>
</tr>
<tr>
<td>first_name</td>
<td>givenName</td>
</tr>
<tr>
<td>last_name</td>
<td>sn</td>
</tr>
<tr>
<td>title</td>
<td>title</td>
</tr>
<tr>
<td>department</td>
<td>department</td>
</tr>
<tr>
<td>manager</td>
<td>manager</td>
</tr>
<tr>
<td>middle_name</td>
<td>initials</td>
</tr>
<tr>
<td>u_memberof</td>
<td>groups</td>
</tr>
<tr>
<td>u_member</td>
<td>members</td>
</tr>
<tr>
<td>u_manager</td>
<td>manager</td>
</tr>
</tbody>
</table>

**LDAP import default mapping**

If you do not specify a transform map (such as LDAP User Import), the integration uses the following default mappings.

### Table 438: LDAP import default mapping

<table>
<thead>
<tr>
<th>User field or variable</th>
<th>LDAP attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name</td>
<td>sAMAccountName</td>
</tr>
<tr>
<td>email</td>
<td>mail</td>
</tr>
<tr>
<td>phone</td>
<td>telephoneNumber</td>
</tr>
<tr>
<td>home_phone</td>
<td>homePhone</td>
</tr>
<tr>
<td>mobile_phone</td>
<td>mobile</td>
</tr>
<tr>
<td>first_name</td>
<td>givenName</td>
</tr>
<tr>
<td>last_name</td>
<td>sn</td>
</tr>
<tr>
<td>title</td>
<td>title</td>
</tr>
<tr>
<td>department</td>
<td>department</td>
</tr>
<tr>
<td>manager</td>
<td>manager</td>
</tr>
<tr>
<td>middle_name</td>
<td>initials</td>
</tr>
<tr>
<td>u_memberof</td>
<td>groups</td>
</tr>
<tr>
<td>u_member</td>
<td>members</td>
</tr>
<tr>
<td>u_manager</td>
<td>manager</td>
</tr>
</tbody>
</table>
**LDAP scripting**

These sample scripts automate common LDAP tasks.

Set disabled Active Directory users to inactive

Use the following script to automatically deactivate users when the associated AD user is disabled.

**Role required:** admin

You can identify disabled Active Directory users by checking the value of the `userAccountControl` attribute. This rule executes whenever the `userAccountControl` value changes and deactivates user accounts if the **User Account Control** signifies a disabled AD account.

Use the following script to automatically deactivate users when the associated AD user is disabled.

1. Configure the User form and create a new integer field called **User Account Control**.
2. Add mapping for `userAccountControl (external)` to the new field.
3. Create a new business rule with the following properties:

<table>
<thead>
<tr>
<th>Business rule field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Disable AD Users</td>
</tr>
<tr>
<td>Table</td>
<td>User [sys_user]</td>
</tr>
<tr>
<td>When</td>
<td>Before</td>
</tr>
<tr>
<td>Condition</td>
<td><code>current.u_user_account_control.changes()</code></td>
</tr>
</tbody>
</table>

The Script field should contain the following:

```javascript
var disabledFlag = 2;
//perform a bitwise comparison on userAccountControl to see if the 2 bit flag is enabled
if (current.u_user_account_control & disabledFlag) {
  gs.log('Disabling user: ' + current.user_name + 'userAccountControl=' +
           current.u_user_account_control);
  current.active='false';
  current.locked_out='true';
}
```

Assign field values for LDAP

You can use a script to assign a value to any field for which there is a field mapping.

For example, to assign a value to the `sys_user.company` field, create a field map for the company field and add a transform script of:

```javascript
company = "Don's Sporting Goods";
```

Exclude particular LDAP users

If you cannot completely filter the LDAP user list using LDAP filter properties, you can exclude users with a map script.
After you have run the logic to identify a user that should not be imported, set the user_name field to an empty string and this user will not be imported.

```
user_name='';
```

One way to identify users to filter out is to look for a string in the distinguishedName attribute. For example, this script excludes accounts that are not in a Users OU. You might use this script if you have too many Users OU to include in the target OU LDAP Option.

```javascript
//vdn is a variable mapped to distinguishedName
gs.include("LDAPUtils");
var vdn = source.getElement(this.distinguishedName);
if (vdn.indexOf('OU=Users')<0) {
    user_name='';
    gs.log('LDAP Import Skipping User: ' + vdn);
}
```

A more complex method of filtering is to use regular expressions.

```javascript
//vcn is a variable mapped to cn
//vdn is a variable mapped to distinguishedName
//c is the regular expression string
gs.include("LDAPUtils");
var vdn = source.getElement(this.distinguishedName);
var vcn = source.getElement(this.cn);
var c = /^[a-z][a-z][a-z][0-9][0-9][0-9]$/;
var nvcn = vcn.toLowerCase();
//test to see if the cn is in the form of 3 letters followed by 3 numbers, only import these
if (c.test(nvcn)) {
    user_name = nvcn;
} else {
    gs.log("LDAP import rejected username: " + vcn + " for DN: " + vdn);
    user_name = "";
}
```

**Verify LDAP mapping**

After creating an LDAP transform map, refresh the LDAP data to verify the transform map works as expected.

**Role required:** admin

1. Navigate to **System LDAP > Scheduled Loads**.
2. Click your LDAP import job.
3. Click **Execute Now**.

**Record creation options during an LDAP transform**

Administrators can specify when to create new records based on changes from incoming LDAP records.

If the LDAP transform map updates a field in the import set table, the integration automatically creates a new record whenever there is a new record in the LDAP data. If the LDAP transform map updates a reference field storing data from another table, the administrator can choose to create, ignore, or reject new LDAP records.

For example, if the integration receives a new department record that does not match any existing department, you may want to update all of the other LDAP record fields without creating a new department record in the instance. The transform map allows you to set the record creation options for each reference field.

Set choice action in the LDAP transform map
The LDAP transform map determines how fields in the Import Set table map to fields in existing tables such as Incident or User.

Role required: admin

1. Navigate to System LDAP > Transform Maps.
2. Select one of the following actions from the Choice action field:
   - **create** – creates a new reference field record if a matching record does not exist.
   - **ignore** – ignores new records in the reference field and completes processing of all other fields in the transform map.
   - **reject** – stops the transform for the entire record.

   **Note:** The field map only displays the Choice action field for reference fields.

## LDAP integration via MID Server

Administrators can integrate using an LDAP data source over a Management, Instrumentation, and Discovery (MID) Server.

The MID Server facilitates communication and movement of data between the platform and external applications, data sources, and services. The MID Server is installed automatically for new instances.

You can use the MID Server to import data over LDAP, but you cannot use the MID Server for LDAP authentication. A MID Server does not support SSL connections.

Using a MID Server to establish an LDAP connection prevents you from having to expose the LDAP server to external network traffic. It also eliminates the need to establish a VPN tunnel between your LDAP server and data centers.

**Note:**
- The MID Server user must have the user_admin role in order to be able to read LDAP server configuration records.
- The following are not available with the MID Server:
  - LDAP authentication
  - SSL connection

### Set up LDAP integration via MID Server

Setting up an LDAP integration via MID Server involves several steps.

Role required: admin

1. Define the LDAP server
2. Configure the LDAP server
   - Provide LDAP server login credentials.
   - Enable a listener.
   - Specify attributes for improving performance or security.
3. Set up communication between the LDAP and MID Servers
4. Define organizational units (OUs) on the LDAP server
5. Create a data source
   - Create a transform map or select an existing transform map.
   - Transform LDAP data into data types that the instance supports.
• Include *onStart and onAfter scripts* in a custom transform map.

6. Create and execute a scheduled data import.
7. Test an LDAP connection manually on page 1837.

Configure an LDAP server
After an LDAP server record exists, you can configure login credentials, enable a listener, and add attributes to the LDAP server that can be used to improve server performance and security.

Role required: admin
1. Provide LDAP server login credentials.
2. Enable a listener.

Provide LDAP server login credentials
The LDAP login credentials determine what organizational units the integration can see.

Role required: admin

Servers that allow anonymous login generally limit the organizational unit (OU) data available to anonymous connections.

1. Navigate to System LDAP > LDAP Servers.
2. Select the LDAP server to configure.
3. In Login distinguished name, enter the user credentials for an account with read access to the directory levels from which you want to import users or groups. The system uses these credentials to connect to your LDAP server. If this information is not entered, the application attempts an anonymous login to the LDAP server.

The Login distinguished name fields accepts several formats.

To access a Microsoft Active Directory (AD) server, use one of the following:

• user@domain.com, domain\user
• cn=user,ou=users,dc=domain,dc=com

To access a different LDAP directory server, the username must be in the full distinguished name format:

• cn=user,ou=users,dc=domain,dc=com

4. In Login password, enter the password for the LDAP user.

   **Note:** Consider enabling LDAPS to encrypt this password during transmission.

5. Select the Active check box.
6. [Optional] In the Starting search directory field, explicitly specify the LDAP OU attributes you want the instance to import.
7. Click Update.

   **Note:**
   If you provide an LDAP password, the integration performs a Simple Bind operation. If you do not provide an LDAP password, the LDAP server must allow anonymous login or the integration cannot bind to the LDAP server.

Specify LDAP attributes
By configuring the LDAP Server form and adding the **Attributes** field, you can specify, and thereby limit, the attributes the LDAP server query returns. This can enhance performance as well as security.

Role required: admin

By default, the system loads all of the attributes for each object that it has permission to read from your LDAP server. By configuring the LDAP Server form and adding the **Attributes** field, you can specify, and thereby limit, the attributes the LDAP server query returns. Using this approach for large LDAP imports can greatly improve the speed of those imports.

For best results, define attributes where possible. If there is information that you do not want exposed to the system, exclude the attribute.

If you do not specify LDAP server attributes, user transactions may freeze for extended periods of time when new attributes are added to an LDAP server object because the system will be busy loading data from the new attributes.

**Note:** To use the manager lookup scripts described in Select or Create a Transform Map for LDAP Data, specify **manager** and **dn** (distinguished name) in the **Attributes** field. Neither attribute is required to be a part of a transform map.
Set up communication between LDAP and MID servers

Configure your LDAP and MID servers to communicate with each other.

Role required: admin

A MID Server connection communicates over HTTP on port 80 by default. This communication channel does not require a certificate. The connection between the MID Server and the instance is over HTTPS (port 443).

An instance can connect to an LDAP server via the MID Server. When you do this, the instance communicates with the MID Server via HTTPS, and the MID Server communicates with the LDAP server via LDAP (port 389). The instance can also connect to the LDAP server directly, using LDAP or LDAPS, either over the internet or through a VPN tunnel.

**Note:** LDAP cannot communicate via the MID Server with password authentication.

To set connection properties for a specific LDAP server:

1. Navigate to **System LDAP > LDAP Servers**.
2. Select the LDAP server to configure.
3. Set the connection property fields (see table).
4. Click Update.

Table 440: LDAP connection properties

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the server.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box if the server is active.</td>
</tr>
<tr>
<td>LDAP Server URLs</td>
<td>Enter the URLs of the primary and backup LDAP servers. Servers are first ordered by operational status, with servers that are Up listed first, then ordered by the Order value that you specify. The first server listed is the primary LDAP server. The others are redundant servers.</td>
</tr>
<tr>
<td>Server URL</td>
<td>Enter the URL of the server. Configure the form to add this field if necessary. It is a calculated read-only field that shows the list of LDAP servers that you can also see in the LDAP Server URLs field, separated by a space, and ordered by operational status and the order values of the URLs.</td>
</tr>
<tr>
<td>Login distinguished name</td>
<td>Enter the distinguished name (DN) of the user authenticating the LDAP connection.</td>
</tr>
<tr>
<td>Login password</td>
<td>Enter the server’s password.</td>
</tr>
<tr>
<td>Starting search directory</td>
<td>Enter the relative distinguished name (RDN) of the default search directory. All queries to this LDAP server will start from this RDN.</td>
</tr>
<tr>
<td>MID Server</td>
<td>Select the MID Server you want to use to connect to the LDAP server. Using a MID Server to establish an LDAP connection prevents you from having to expose the LDAP server to external network traffic. It also eliminates the need to establish a VPN tunnel between your LDAP server and ServiceNow data centers.</td>
</tr>
</tbody>
</table>

Note:
- The MID Server user must have the user_admin role in order to be able to read LDAP server configuration records.
- The following are not available with the MID Server:
  - LDAP authentication
  - SSL connection
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect timeout</td>
<td>Specify the maximum number of seconds that the instance has to establish an LDAP connection. If no connection is made by this time, the connection is terminated.</td>
</tr>
<tr>
<td>Read timeout</td>
<td>Specify the number of seconds the integration has to read LDAP data. The integration stops reading LDAP data after the connection exceeds the read timeout. If you enable an SSL connection, you can also set a read timeout value with the com.glide.ssl.read.timeout system property. If you enter timeout values for both this field and the system property, the lowest timeout value takes precedence.</td>
</tr>
<tr>
<td>SSL</td>
<td>Select this check box to require the LDAP server to make an SSL-encrypted connection. For more information, see Enable SSL. If you selected a MID Server, this field is not available.</td>
</tr>
<tr>
<td>Listener</td>
<td>Select this check box to enable the integration to periodically poll Microsoft Active Directory servers or LDAP servers that support persistent search request control. Additionally, if you selected a MID Server, the listener functionality is available for that MID Server.</td>
</tr>
<tr>
<td>Listen interval (timeout value)</td>
<td>Specify the listener timeout value in the number of minutes that the integration listens for LDAP data with every connection. The integration stops listening for LDAP data after the connection exceeds the listen interval.</td>
</tr>
<tr>
<td>Paging</td>
<td>Select this check box to have the LDAP server split up LDAP attribute data into multiple result sets rather than submit multiple queries.</td>
</tr>
</tbody>
</table>

**Test communication between LDAP and MID Servers**

The instance tests the connection automatically every time a user opens the LDAP Server form.

Role required: admin
The instance supports an LDAP connection timeout of 29 seconds or less. Error messages appear on the form if there are any issues connecting to the LDAP server.

Employees can also verify connectivity between the instance and the LDAP server. Contact Technical Support for assistance verifying LDAP connectivity.

Configure LDAP connection monitoring
Change or disable LDAP connection monitoring and notifications.

Role required: admin

The instance automatically sends an email to users configured in the LDAP Admins group when an LDAP server connection fails. This uses the email notification, which is launched by the LDAP Connection Test scheduled job. This email notification is enabled by default.

Note: The instance does not send the email notification unless there is at least one member in the LDAP Admins group. Make sure to populate this group with the users you want to receive the email.

By default, the scheduled job tests the connection every 15 minutes. To change this interval or disable monitoring:

1. Navigate to System Definition > Scheduled Jobs.
2. Open LDAP Connection Test.
3. Do one of the following:
   • Change the interval in the Repeat Interval field.
   • Disable monitoring by clearing the Active check box.

Scheduled data imports for LDAP
A scheduled import allows administrators to import LDAP data on a regular schedule.

By default, the LDAP integration includes two sample scheduled imports:

• Example LDAP User Import
• Example LDAP Group Import

Neither example is active by default. Change these scheduled imports to meet your company's business needs.

Import binary data through a MID Server
As an administrator, you can import binary large object (BLOB) data with an LDAP integration through the MID Server.

Role required: admin

Use the examples in the following steps to add choice tables, notes, tables, figures, step results, and postrequisites to the task.

1. Add the name of the LDAP column you want to import binary data from to the system property glide.ldap.binary_attributes.
2. Add a MID Server property with the Name glide.ldap.binary_attributes and the same value you set for the system property.

Create a concept topic to introduce the background needed to perform this process or task.

Troubleshooting LDAP integration via MID Server
You may encounter issues in the following areas while integrating LDAP via MID Server.

You can troubleshoot these issues by viewing the outputs found in the External Communication Channel (ECC) Queue (Discovery > Output and Artifacts > ECC Queue).
Test Connection Issues

When defining OUs within the server, there is a Test connection related list that is used to verify the LDAP connection. When you click this link, the ECC Queue should show a single output message with a topic name of LDAPConnectionTesterProbe. After the test has completed on the MID Server, the ECC Queue should show an input message with the same topic name. If the Name column for the input message shows true, the test was successful. Drill down into the record to view the payload and ensure it does not contain error messages.

![Figure 518: Test Connection](image)

Browse Issues

When defining OUs within the server, there is a Browse related list that is used to view the LDAP directory records that the OU definition returns. When you click this link, the ECC Queue should show a single output message with a topic name of LDAPBrowseProbe. After data has been returned from the MID Server, the ECC Queue should show an input message with the same topic name. If the Name column for the input message shows true, the test was successful. Drill down into the record to view the payload and ensure it does not contain error messages.

Load Import Issues

When uploading data (for example, using the Test Load 20 Records feature), the ECC Queue should show a single output message with a topic name of LDAPProbe. After data has been returned from the MID Server, the ECC Queue should show another input message called LDAPProbeCompleted. The Name column for this input message shows the total number of records returned.

An additional input messages, also named LDAPProbe, is displayed. The Name column for this input message displays the highest record number in the batch. If the total number of records returned is 258 and the batch size is 200 (the default), two LDAPProbe (200, 258) incoming messages will be received, and one LDAPProbeCompleted (258) incoming message will be received.

Drill down into the record to view the payload and ensure it does not contain error messages.

![Figure 519: Import Load](image)

Also keep an eye out for an output message called LDAPProbeError.
Figure 520: Error message

Click the link in the Name column to view the details of the error.

LDAP integration troubleshooting

If you are integrating your LDAP server and have questions, these items may help you troubleshoot the issue.

Preliminary checks

- If the LDAP is unavailable, users cannot log in to the instance. A good practice is to have local accounts for administrators so that if the LDAP is down, administrators can still access the instance.
- Check the service account to ensure that it is not expired or locked out.
- Check the format of the user name. Instead of using just the user name, try using the domain with the user name, or username@domain.
- Verify that you have changed the system_id entry on the ldap_server_config record. If you modify the system_id unintentionally with an Update Set, system_id points to the wrong node for the target instance and does not work.

Error codes

The LDAP Log file lists industry standard error codes for both LDAP and Active Directory (AD). The LDAP error codes are two-digit numbers, while the Active Directory error codes are three-digit numbers. For a list of the most-common error codes, see LDAP Error Codes.

Multiple domain integration

You can integrate multiple domains within the same forest or in completely non-trusted domains. It is recommended that you create a separate LDAP server record for each domain. Each LDAP server record must point to a domain controller for that given domain. This means you will have to allow connections to each of the domain controllers. Multiple AD forests through LDAP with one LDAP account is not supported.

When you expand to more than one domain, it is critical that you identify unique LDAP attributes for the application user names and import coalesce values. A common unique coalesce attribute for Active Directory is objectSid. Unique user names will vary based on your LDAP data design. Common unique attributes are email or userPrincipalName.

Incoming records

See Record creation options during an LDAP transform on page 1849 to set how the integration processes incoming LDAP records that are missing matching values in reference fields.
Common authentication errors

- User Cannot Log In (Invalid DN)
- Invalid CN
- Invalid Connection

Automatic LDAP connection tests

You can manually test connections to LDAP servers or allow ServiceNow to automatically test the connections.

The system tests the connection automatically:

- Every time a user opens the LDAP Server form
- Through the LDAP Connection Test scheduled job, which runs every 15 minutes by default.

You can change how often this scheduled job runs. If this scheduled job is not able to establish a connection, a new one-time schedule job retries the connection test after either five minutes, or half the Repeat Interval value in the scheduled job, whichever occurs first.

Error messages appear on the form if there are any issues connecting to the LDAP server. Also supported are test connections for servers behind a MID server.

Troubleshooting LDAP integration via MID Server

You may encounter issues in the following areas while integrating LDAP via MID Server.

You can troubleshoot these issues by viewing the outputs found in the External Communication Channel (ECC) Queue (Discovery > Output and Artifacts > ECC Queue).

Test Connection Issues

When defining OUs within the server, there is a Test connection related list that is used to verify the LDAP connection. When you click this link, the ECC Queue should show a single output message with a topic name of LDAPConnectionTesterProbe. After the test has completed on the MID Server, the ECC Queue should show an input message with the same topic name. If the Name column for the input message shows true, the test was successful. Drill down into the record to view the payload and ensure it does not contain error messages.

Browse Issues

When defining OUs within the server, there is a Browse related list that is used to view the LDAP directory records that the OU definition returns. When you click this link, the ECC Queue should show a single output message with a topic name of LDAPBrowseProbe. After data has been returned from the MID Server, the ECC Queue should show an input message with the same topic name. If the Name column for the input message shows true, the test was successful. Drill down into the record to view the payload and ensure it does not contain error messages.
Load Import Issues

When uploading data (for example, using the Test Load 20 Records feature), the ECC Queue should show a single output message with a topic name of **LDAPProbe**.

After data has been returned from the MID Server, the ECC Queue should show another input message called **LDAPProbeCompleted**. The **Name** column for this input message shows the total number of records returned.

An additional input messages, also named **LDAPProbe**, is displayed. The **Name** column for this input message displays the highest record number in the batch. If the total number of records returned is 258 and the batch size is 200 (the default), two LDAPProbe (200, 258) incoming messages will be received, and one LDAPProbeCompleted (258) incoming message will be received.

Drill down into the record to view the payload and ensure it does not contain error messages.

![Figure 522: Import Load](image)

Also keep an eye out for an output message called **LDAPProbeError**.

![Figure 523: Error message](image)

Click the link in the **Name** column to view the details of the error.

**Test an LDAP connection manually**

You can manually test the connection to the LDAP server from the LDAP server form.

Role required: admin

1. Navigate to **System LDAP > LDAP Servers**.
2. Select the LDAP server to test.
3. Under **Related Links**, click **Test connection**.
4. Under **Related Links**, click **Browse** to verify that the appropriate LDAP directory structure is visible to the system.
5. (Optional) If the connection was successful, click **Browse** to view the source LDAP directory structure that is visible to the instance.

**Note:**

The **Filter** and **RDN** fields on the left of the Browse window are ignored when you use the search field on the right.

**LDAP error codes**

The LDAP Log file lists industry standard error codes for both LDAP and Active Directory (AD).
### Standard error codes

#### Table 441: Standard LDAP errors

<table>
<thead>
<tr>
<th>Error / data code</th>
<th>Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LDAP_SUCCESS</td>
<td>Indicates the requested client operation completed successfully.</td>
</tr>
<tr>
<td>2</td>
<td>LDAP_PROTOCOL_ERROR</td>
<td>Indicates that the server has received an invalid or malformed request from the client.</td>
</tr>
<tr>
<td>3</td>
<td>LDAP_TIMELIMIT_EXCEEDED</td>
<td>Indicates that the operation's time limit specified by either the client or the server has been exceeded. On search operations, incomplete results are returned.</td>
</tr>
<tr>
<td>4</td>
<td>LDAP_SIZELIMIT_EXCEEDED</td>
<td>Indicates that in a search operation, the size limit specified by the client or the server has been exceeded. Incomplete results are returned.</td>
</tr>
<tr>
<td>5</td>
<td>LDAP_COMPARE_FALSE</td>
<td>Does not indicate an error condition. Indicates that the results of a compare operation are false.</td>
</tr>
<tr>
<td>6</td>
<td>LDAP_COMPARE_TRUE</td>
<td>Does not indicate an error condition. Indicates that the results of a compare operation are true.</td>
</tr>
<tr>
<td>7</td>
<td>LDAP_AUTH_METHOD_NOT_SUPPORTED</td>
<td>Indicates that during a bind operation the client requested an authentication method not supported by the LDAP server.</td>
</tr>
<tr>
<td>8</td>
<td>LDAP_STRONG_AUTH_REQUIRED</td>
<td>Indicates one of the following: In bind requests, the LDAP server accepts only strong authentication. In a client request, the client requested an operation such as delete that requires strong authentication. In an unsolicited notice of disconnection, the LDAP server discovers the security protecting the communication between the client and server has unexpectedly failed or been compromised.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Error / data code</th>
<th>Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Reserved.</td>
<td>Reserved.</td>
</tr>
<tr>
<td>10</td>
<td>LDAP_REFERRAL</td>
<td>Does not indicate an error condition. In LDAPv3, indicates that the server does not hold the target entry of the request, but that the servers in the referral field may.</td>
</tr>
<tr>
<td>11</td>
<td>LDAP_ADMINLIMIT_EXCEEDED</td>
<td>Indicates that an LDAP server limit set by an administrative authority has been exceeded.</td>
</tr>
<tr>
<td>12</td>
<td>LDAP_UNAVAILABLE_CRITICAL_EXTENSION</td>
<td>Indicates that the LDAP server was unable to satisfy a request because one or more critical extensions were not available. Either the server does not support the control or the control is not appropriate for the operation type.</td>
</tr>
<tr>
<td>13</td>
<td>LDAP_CONFIDENTIALITY_REQUIRED</td>
<td>Indicates that the session is not protected by a protocol such as Transport Layer Security (TLS), which provides session confidentiality.</td>
</tr>
<tr>
<td>14</td>
<td>LDAP_SASL_BIND_IN_PROGRESS</td>
<td>Does not indicate an error condition, but indicates that the server is ready for the next step in the process. The client must send the server the same SASL mechanism to continue the process.</td>
</tr>
<tr>
<td>15</td>
<td>Not used.</td>
<td>Not used.</td>
</tr>
<tr>
<td>16</td>
<td>LDAP_NO_SUCH_ATTRIBUTE</td>
<td>Indicates that the attribute specified in the modify or compare operation does not exist in the entry.</td>
</tr>
<tr>
<td>17</td>
<td>LDAP_UNDEFINED_TYPE</td>
<td>Indicates that the attribute specified in the modify or add operation does not exist in the LDAP server's schema.</td>
</tr>
<tr>
<td>18</td>
<td>LDAP_INAPPROPRIATE_MATCHING</td>
<td>Indicates that the matching rule specified in the search filter does not match a rule defined for the attribute's syntax.</td>
</tr>
<tr>
<td>Error / data code</td>
<td>Text</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>19</td>
<td>LDAP_CONSTRAINT_VIOLATION</td>
<td>Indicates that the attribute value specified in a modify, add, or modify DN operation violates constraints placed on the attribute. The constraint can be one of size or content (string only, no binary).</td>
</tr>
<tr>
<td>20</td>
<td>LDAP_TYPE_OR_VALUE_EXISTS</td>
<td>Indicates that the attribute value specified in a modify or add operation already exists as a value for that attribute.</td>
</tr>
<tr>
<td>21</td>
<td>LDAP_INVALID_SYNTAX</td>
<td>Indicates that the attribute value specified in an add, compare, or modify operation is an unrecognized or invalid syntax for the attribute.</td>
</tr>
<tr>
<td>22-31</td>
<td>Not used.</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>LDAP_NO_SUCH_OBJECT</td>
<td>Indicates the target object cannot be found. This code is not returned on following operations: Search operations that find the search base but cannot find any entries that match the search filter. Bind operations.</td>
</tr>
<tr>
<td>33</td>
<td>LDAP_ALIAS_PROBLEM</td>
<td>Indicates that an error occurred when an alias was dereferenced.</td>
</tr>
<tr>
<td>34</td>
<td>LDAP_INVALID_DN_SYNTAX</td>
<td>Indicates that the syntax of the DN is incorrect. (If the DN syntax is correct, but the LDAP server’s structure rules do not permit the operation, the server returns code 53: LDAP_UNWILLING_TO_PERFORM.)</td>
</tr>
<tr>
<td>35</td>
<td>LDAP_IS_LEAF</td>
<td>Indicates that the specified operation cannot be performed on a leaf entry. (This code is not currently in the LDAP specifications, but is reserved for this constant.)</td>
</tr>
<tr>
<td>36</td>
<td>LDAP_ALIAS_DEREF_PROBLEM</td>
<td>Indicates that during a search operation, either the client does not have access rights to read the aliased object’s name or dereferencing is not allowed.</td>
</tr>
<tr>
<td>37-47</td>
<td>Not used.</td>
<td></td>
</tr>
<tr>
<td>Error / data code</td>
<td>Text</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>48</td>
<td>LDAP_INAPPROPRIATE_AUTH</td>
<td>Indicates that during a bind operation, the client is attempting to use an authentication method that the client cannot use correctly. For example, either of the following cause this error: The client returns simple credentials when strong credentials are required...OR...The client returns a DN and a password for a simple bind when the entry does not have a password defined.</td>
</tr>
<tr>
<td>49</td>
<td>LDAP_INVALID_CREDENTIALS</td>
<td>Indicates that during a bind operation one of the following occurred: The client passed either an incorrect DN or password, or the password is incorrect because it has expired, intruder detection has locked the account, or another similar reason. See the data code for more information.</td>
</tr>
<tr>
<td>49 / 52e</td>
<td>AD_INVALID CREDENTIALS</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext error, which is returned when the username is valid but the combination of password and user credential is invalid. This is the AD equivalent of LDAP error code 49.</td>
</tr>
<tr>
<td>49 / 525</td>
<td>USER NOT FOUND</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext data error that is returned when the username is invalid.</td>
</tr>
<tr>
<td>49 / 530</td>
<td>NOT_PERMITTED_TOLOGIN</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext data error that is logon failure caused because the user is not permitted to log on at this time. Returns only when presented with a valid username and valid password credential.</td>
</tr>
<tr>
<td>Error / data code</td>
<td>Text</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>49 / 531</td>
<td>RESTRICTED_TO_SPECIFIC_MACHINES</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext data error that is logon failure caused because the user is not permitted to log on from this computer. Returns only when presented with a valid username and valid password credential.</td>
</tr>
<tr>
<td>49 / 532</td>
<td>PASSWORD_EXPIRED</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext data error that is a logon failure. The specified account password has expired. Returns only when presented with valid username and password credential.</td>
</tr>
<tr>
<td>49 / 533</td>
<td>ACCOUNT_DISABLED</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext data error that is a logon failure. The account is currently disabled. Returns only when presented with valid username and password credential.</td>
</tr>
<tr>
<td>49 / 568</td>
<td>ERROR_TOO_MANY_CONTEXT_IDS</td>
<td>Indicates that during a log-on attempt, the user's security context accumulated too many security IDs. This is an issue with the specific LDAP user object/account which should be investigated by the LDAP administrator.</td>
</tr>
<tr>
<td>49 / 701</td>
<td>ACCOUNT_EXPIRED</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext data error that is a logon failure. The user's account has expired. Returns only when presented with valid username and password credential.</td>
</tr>
<tr>
<td>49 / 773</td>
<td>USER MUST RESET PASSWORD</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext data error. The user's password must be changed before logging on the first time. Returns only when presented with valid user-name and password credential.</td>
</tr>
<tr>
<td>Error / data code</td>
<td>Text</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>50</td>
<td>LDAP_INSUFFICIENT_ACCESS</td>
<td>Indicates that the caller does not have sufficient rights to perform the requested operation.</td>
</tr>
<tr>
<td>51</td>
<td>LDAP_BUSY</td>
<td>Indicates that the LDAP server is too busy to process the client request at this time but if the client waits and resubmits the request, the server may be able to process it then.</td>
</tr>
<tr>
<td>52</td>
<td>LDAP_UNAVAILABLE</td>
<td>Indicates that the LDAP server cannot process the client's bind request, usually because it is shutting down.</td>
</tr>
<tr>
<td>52e</td>
<td>AD_INVALID_CREDENTIALS</td>
<td>Indicates an Active Directory (AD) AcceptSecurityContext error, which is returned when the username is valid but the combination of password and user credential is invalid. This is the AD equivalent of LDAP error code 49: LDAP_INVALID_CREDENTIALS.</td>
</tr>
<tr>
<td>53</td>
<td>LDAP_UNWILLING_TO_PERFORM</td>
<td>Indicates that the LDAP server cannot process the request because of server-defined restrictions. This error is returned for the following reasons: The add entry request violates the server's structure rules...OR...The modify attribute request specifies attributes that users cannot modify...OR...Password restrictions prevent the action...OR...Connection restrictions prevent the action.</td>
</tr>
<tr>
<td>54</td>
<td>LDAP_LOOP_DETECT</td>
<td>Indicates that the client discovered an alias or referral loop, and is thus unable to complete this request.</td>
</tr>
<tr>
<td>55-63</td>
<td></td>
<td>Not used.</td>
</tr>
<tr>
<td>Error / data code</td>
<td>Text</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>64</td>
<td>LDAP_NAMING_VIOLATION</td>
<td>Indicates that the add or modify DN operation violates the schema's structure rules. For example, The request places the entry subordinate to an alias. The request places the entry subordinate to a container that is forbidden by the containment rules. The RDN for the entry uses a forbidden attribute type.</td>
</tr>
<tr>
<td>65</td>
<td>LDAP_OBJECT_CLASS_VIOLATION</td>
<td>Indicates that the add, modify, or modify DN operation violates the object class rules for the entry. For example, the following types of request return this error: The add or modify operation tries to add an entry without a value for a required attribute. The add or modify operation tries to add an entry with a value for an attribute which the class definition does not contain. The modify operation tries to remove a required attribute without removing the auxiliary class that defines the attribute as required.</td>
</tr>
<tr>
<td>66</td>
<td>LDAP_NOT_ALLOWED_ON_NONLEAF</td>
<td>Indicates that the requested operation is permitted only on leaf entries. For example, the following types of requests return this error: The client requests a delete operation on a parent entry. The client request a modify DN operation on a parent entry.</td>
</tr>
<tr>
<td>67</td>
<td>LDAP_NOT_ALLOWED_ON_RDN</td>
<td>Indicates that the modify operation attempted to remove an attribute value that forms the entry’s relative distinguished name.</td>
</tr>
<tr>
<td>68</td>
<td>LDAP_ALREADY_EXISTS</td>
<td>Indicates that the add operation attempted to add an entry that already exists, or that the modify operation attempted to rename an entry to the name of an entry that already exists.</td>
</tr>
<tr>
<td>Error / data code</td>
<td>Text</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>69</td>
<td>LDAP_NO_OBJECT_CLASS_MODS</td>
<td>Indicates that the modify operation attempted to modify the structure rules of an object class.</td>
</tr>
<tr>
<td>70</td>
<td>LDAP_RESULTS_TOO_LARGE</td>
<td>Reserved for CLDAP.</td>
</tr>
<tr>
<td>71</td>
<td>LDAP_AFFECTS_MULTIPLE_DNAS</td>
<td>Indicates that the modify DN operation moves the entry from one LDAP server to another and requires more than one LDAP server.</td>
</tr>
<tr>
<td>72-79</td>
<td></td>
<td>Not used.</td>
</tr>
<tr>
<td>80</td>
<td>LDAP_OTHER</td>
<td>Indicates an unknown error condition. This is the default value for NDS error codes which do not map to other LDAP error codes.</td>
</tr>
</tbody>
</table>

**Customized error codes**

Table 442: Customized LDAP error codes

<table>
<thead>
<tr>
<th>Error / data code</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>LDAP_ERROR_GENEREL</td>
</tr>
<tr>
<td>10001</td>
<td>LDAP_ERROR_MAL_FORMED_URL</td>
</tr>
<tr>
<td>10002</td>
<td>LDAP_ERROR_UNAUTHENTICATED_BIND</td>
</tr>
<tr>
<td>10300</td>
<td>LDAP_ERROR_COMMUNICATION_EXCEPTION</td>
</tr>
<tr>
<td>10301</td>
<td>LDAP_ERROR_SOCKET_TIMEOUT</td>
</tr>
<tr>
<td>10302</td>
<td>LDAP_ERROR_CONNECTION_REFUSED</td>
</tr>
<tr>
<td>10303</td>
<td>LDAP_ERROR_CONNECTION_RESET</td>
</tr>
<tr>
<td>10304</td>
<td>LDAP_ERROR_NO_ROUTE</td>
</tr>
<tr>
<td>10305</td>
<td>LDAP_ERROR_UNKNOW_HOST</td>
</tr>
<tr>
<td>10400</td>
<td>LDAP_ERROR_SSL_EXCEPTION</td>
</tr>
<tr>
<td>10401</td>
<td>LDAP_ERROR_SSL_EMPTY_CERT_STORE</td>
</tr>
<tr>
<td>10402</td>
<td>LDAP_ERROR_SSL_CERT_NOT_FOUND</td>
</tr>
<tr>
<td>10403</td>
<td>LDAP_ERROR_SSL_CERT_EXPIRED</td>
</tr>
<tr>
<td>10500</td>
<td>LDAP_ERROR_INVALID_SEARCH_FILTER_EXCEPTION</td>
</tr>
</tbody>
</table>

**LDAP record synchronization**

Administrators can synchronize inactive, disabled, or deleted LDAP records with their LDAP records.
LDAP record synchronization is the process of detecting inactive records on the LDAP server and updating the corresponding LDAP records. Detecting inactive LDAP records involves defining consistent data indicators for each user object, importing LDAP data, and evaluating the data indicators.

A data indicator can be:

- a date field
- membership in a specific OU (identify by parsing the dn attribute), using the useraccountcontrol attribute
- a combination of these indicators

Imported data comes into the instance through import set tables where the data can be evaluated and processed.

The import process can use:

- **LDAP extraction** on page 1870: a single import job to gather all user records into the import set temporary tables for evaluation
- **LDAP refresh filters** on page 1869: multiple import jobs to divide different types of user records, segregating records for separate processing

**LDAP refresh filters**

Filters on the LDAP refresh process can be used to specify processing that ignores inserts of disabled users.

You can loosen the LDAP OU filter to bring all of the data in to your import set table (including inactive users) and then specify processing that ignores inserts of disabled users. The sample ‘Users’ OU definition that the instance provides in its out-of-box LDAP sample contains a filter.

This filter is important because it defines which user records are brought into the import set table to be evaluated. While achieving a smaller data load, a limitation of this filter is that it filters out inactive users, so the inactive user records are not imported into the import set temporary tables. Since there is not visibility of the inactive user records, there is no ability to evaluate the record indicators.
To use filtering within the main LDAP refresh process, change the filter to bring in all of the user records. The result is that all the records will be loaded into the import set temporary table where they can be evaluated and transformed.

**Note:** There is a precaution here: because the filtering brings in all the records, you may end up with a vast amount of older inactive LDAP accounts that should not be inserted into the instance. A user record should never be created for a disabled user.

**Alternative method**

*LDAP extraction* on page 1870 uses a single import job to gather all user records into the import set temporary tables for evaluation.

*LDAP extraction*  
An LDAP extraction process can be implemented to detect disabled users.

An extract from your LDAP source can filtered for disabled users using an active flag that can be set for every record in the import to ‘false’. Specify (`target.active=false`) and copy into the **Script** field directly on the Table Transform Map record.
Benefits

Benefits to this method include:

• Simple scripting
• Existing user records are not involved in processing
• Inactive users are not loaded into a temporary import table
• No performance impact

Drawbacks

Drawbacks to this method include:

• An additional process is created
• The extract set must be placed in a location where your data source can access it

Alternative method

LDAP refresh filters on page 1869 use multiple import jobs to divide different types of user records, segregating records for separate processing.

Inactive LDAP user accounts
Detect that an existing, current, user account is inactive or has been disabled or deleted from an Active Directory (AD) LDAP.

A common LDAP integration issue is how to detect disabled or deleted users in an Active Directory (AD) and then deactivate them in the instance. In an Active Directory LDAP, a filter is usually set to exclude inactive users when refreshing, so the instance is not aware of users that are disabled or deleted in AD.

The issue is how to detect that an existing, current user is inactive or has been deleted from AD.

Note: The recommended approach is to deactivate user records and all other types of records, not delete them. Each record is linked to other records, and deleting a record destroys all the relationships to those other records. Deactivating records keeps those relationships in place.

There are two approaches that you can use to find disabled and deleted AD accounts to synchronize your user records:

• Find inactive LDAP accounts using the lastRefresh time on page 1871
• Find inactive LDAP accounts using the userAccountControl field on page 1872

Find inactive LDAP accounts using the lastRefresh time
Locate accounts with inactive or missing LDAP connections.

Role required: admin

One method is to add a lastRefresh field to the user record and set the value during the import process. Then create a scheduled job that checks for users that have not been refreshed in 30 days, and deactivate them.

Warning: If the LDAP import fails for 30 days then everyone is deactivated.

To find and deactivate inactive user accounts:

1. Create a datetime field on the User [sys_user] table. For example, u_last_refreshed.
2. Create an LDAP transform script to set the field value.

```javascript
target.u_last_refreshed = gs.now();
```

3. Create a scheduled job to find and deactivate the user accounts that have not been refreshed in 30 days.

```javascript
disable_users();
function disable_users() {
    /* query for active users with ldap source and last updated more than 30 days ago
    * disable them
    */
    var gr = new GlideRecord("sys_user");
gr.addQuery('u_last_refreshed', '<', gs.daysAgoStart(30));
gr.addQuery('active', true);
gr.addQuery('source', '!=', '');
gr.query();
while (gr.next()) {
gr.active = false;
    gs.log("Disabled inactive user: " + gr.user_name + " - last updated: " + u_last_refreshed);
gr.update();
}
gs.log("Completed disabling inactive accounts");
}
```

4. Create a report of user accounts that have been inactive for 15 days.

Find inactive LDAP accounts using the userAccountControl field
Identify when an Active Directory (AD) user is deleted (or made inactive).

Role required: admin

One method is to track the active status of AD users and create a business rule to update corresponding accounts when an AD account is inactive.

To find and deactivate inactive user accounts:

1. Create a new string field on the User [sys_user] table to track the value of the AD userAccountControl field. For example: u_ad_user_account.

2. Create an LDAP transform script to set the field value.

```javascript
target.u_ad_user_account = source.userAccountControl
```

3. Update the LDAP filter to show disabled AD accounts.
Here is an example of a filter.

```javascript
(&(objectClass=person) (sn=*)(!(objectClass=computer))(!(userAccountControl:1.2.840.113556.1.4.803:=2)))
```

Here is an example of a replacement filter you can use.

```javascript
(&(objectClass=person) (sn=*)(!(objectClass=computer)))
```

4. Create an onChange business rule to set the active field to false whenever the u_ad_user_account field has the value 514.

'514' indicates an inactive account.
**LDAP script examples**
The following script examples assume you use an Active Directory (AD) for your LDAP server.

**userAccountControl attribute values script**
This example tests the source for the userAccountControl attribute values associated with a disabled user (514 or 546).

```javascript
//Deactivate LDAP-disabled users during transform based on 'userAccountControl' attribute
if(source.u_useraccountcontrol == '514' || source.u_useraccountcontrol == '546'){
   target.active=false;
   target.locked_out=true;
}
```

Here is an example using a bitwise check:

```javascript
if(source.u_useraccountcontrol & 2){
   active =false;
}
```

**userAccountControl attribute script**
This example examines the userAccountControl attribute but does not test for specific values. It also contains the option of reactivating LDAP user accounts.

```javascript
/*
* Deactivate LDAP-disabled users during transform based on 'userAccountControl' attribute
* Convert the userAccountControl attribute back to a hex value
*/
var ctrl = parseInt(source.u_useraccountcontrol, 10);
ctrl = ctrl.toString(16);

/*. The only digit we care about is the final one
* A final hex digit value of '2' in 'ctrl' means disabled
*/
if(ctrl.substr(-1) == "2"){
   //Deactivate and lock the user account
   target.active = false;
   target.locked_out = true;

   //Ignore any insert of a disabled record
   if(action == 'insert'){
      ignore = true;
   }
}
/* Optional: Uncomment else block to reactivate and unlock the user account
else {
   target.active = true;
   target.locked_out = ctrl.substr(-2, 1) == "1";
}
*/
onBefore transform map script

Here is an example of a onBefore transform map script. The script identifies disabled records and records being inserted. If an insert of a disabled user is occurring, then the operation transform ignores the record.

```javascript
//Ignore any insert of a disabled record as defined by the 'userAccountControl' attribute
var uc = source.u_useraccountcontrol;
if((uc == '514' || uc == '546') && action == 'insert'){
  ignore = true;
}
```

DN member script

This script example introduces flexibility by not relying on the 546 and 514 userAccountControl values, but instead checking whether the user is a member of a particular Distinguished Name (DN). You can use this script either in the Script field of the ‘Table Transform Map’ record or in an onBefore transform map script.

```javascript
//Deactivate LDAP-disabled users during transform based on OU membership in 'dn'
if(source.u_dn.indexOf('OU=Disabled Accounts') > -1){
  target.active = false;
  target.locked_out = true;
}
```

Active Directory Application Mode (ADAM)

Active Directory Application Mode (ADAM) is an Lightweight Directory Access Protocol (LDAP)-compliant directory service.

Note: A basic level of understanding with Microsoft Windows Server and Active Directory is needed for understanding this topic. You must also have administrator permissions on the server you are configuring for ADAM.

These are sample procedures. Due to installation and environment variations, we cannot offer direct support. We recommend working with a Microsoft consultant.

ADAM has a simple install and runs as a service on Windows operating systems. It can be fully customized and distributed as an application component or used as a stand-alone LDAP directory. ADAM uses the same technologies found on Active Directory Domain Controllers (including replication and delegation features) and has its own administration and customization features. It can be run as a Windows service. ADAM can be installed on Windows XP, 2000, 2003, and 2008 operating systems. ADAM is included as part of Windows Server 2003 R2 and Windows Server 2008. A download is available at [http://www.microsoft.com/downloads](http://www.microsoft.com/downloads) for earlier operating systems.

Security

Some company security policies prohibit external vendors and partners from connecting directly to an Active Directory (AD) Domain Controller. If exposing certain AD objects or attributes to an external vendor or partner is prohibited, access to objects and attributes can be blocked using AD Security Access Control Entries (ACE or ACL). Depending on security requirements, this method can introduce complexity in the integration. Consolidating multiple domains and forests is recommended. If all LDAP imports and authentications need to be channeled through a single source, ADAM can be used as a consolidated source. With the release of Windows 2008 this functionality has been renamed to Light-Weight-Directory Service, LDS. Installation and configuration is similar to Windows Server 2003 R2.
**Recommended Knowledge**

For this task, you must understand AD, object classes and attributes. To have a successful integration, you need to be knowledgeable of the current AD object structure, familiar with Active Directory delegations, and have a strategy on how to use ADAM and for what purposes. If you are not familiar with AD or ADAM, work with your AD administrator to configure a new ADAM environment.

**Trusts**

If `userProxy` objects is used, the computer hosting ADAM needs to be a member of the domain that has the AD accounts, or a member of a trusted domain.

**Internal Connectivity**

If `userProxy` objects is used, the ADAM computer must be able to connect to the related Domain Controllers to perform proxy authentication.

*Configuring an instance with ADAM*

The first install copies the ADAM files to your computer, registers requires components, and creates the application shortcuts.

Role required: admin

By default, all of the application files are installed to `%systemroot%\ADAM`.

- Windows Server 2003 R2 - ADAM can be installed using the Control Panel > Add and Remove Programs > Optional Component Manager.

Create the first instance service which functions as the first directory service hosted by ADAM. Do one of the following:

- Run `adaminstall.exe` from the ADAM folder.
- Use the Create an ADAM instance shortcut from the Start Menu > Programs > ADAM folder.
  a) Select the A unique instance install option.

  **Note:** You can use this option to install an instance replica on a second server to provide a fault tolerant system.

  b) Complete the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Name</td>
<td>used primarily to identify the Windows Service name and display name</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ports</td>
<td>sets the port numbers to be used for LDAP and LDAPS Listeners. The default LDAP port is 389, LDAPS is 636. If these ports are in use on the server, the setup wizard selects new ports. Work with your network administrator to determine the best ports to use.</td>
</tr>
<tr>
<td>Application Directory Partition</td>
<td>creates an application directory partition. Not needed at this step, we recommend creating the new partition now. A good practice is to use the same distinguished name as your forest or domain, but replace the highest level domain with adun instead of com or local. For example, if your forest partition is <code>dc=myCompany,dc=com</code>, you could create the ADAM partition as <code>dc=myCompany,dc=adam</code>.</td>
</tr>
<tr>
<td>File Locations</td>
<td>selects the location(s) for the ADAM partition data.</td>
</tr>
<tr>
<td>Service Account Selection</td>
<td>selects a service account that the instance runs as. For stand-alone services, you can use the default network service account. If you plan on using replicas, you need to use an account that has access to all ADAM instances.</td>
</tr>
<tr>
<td>ADAM Administrators</td>
<td>the delegation on the ADAM directory that leverages Windows integrated authentication. This is how the initial access is granted for administration. Once the initial account is granted rights, this user or group delegates rights to other Windows users or ADAM users. You can select the default to only grant admin access to the current user, or grant access to a different user or group based on your needs.</td>
</tr>
<tr>
<td>Import LDIF Files</td>
<td>the files to import. MS-UserProxy is the most important file to import, but it’s worth adding all available files since there is little overhead to the schema and you won’t have to worry about extending it later if your needs expand. Confirm the details and the wizard complete the configuration.</td>
</tr>
</tbody>
</table>
Setting up the ADAM console

Even though there are many similarities between ADAM and Active Directory, the administration can be very different since there is no Users and Computers management console.

Role required: admin

Most of the general administration is performed using the ADAM ADSI MMC console available from the ADAM start menu. The first time you run the ADAM ADSI console, you must connect to the partition you created.

1. Right-click the ADAM ADSI Edit item in the left frame.
2. Give the new connection a name and update the server name, port fields with the information used when you created the instance.
3. Select distinguished name or naming context and specify the distinguished name of the application partition you created earlier. You can connect to the Configuration and Schema partitions for advanced configuration options.
   You should now be able to see into the partition and the default containers for LostAndFound, NTDS Quotas, and Roles. The Roles container has not been configured yet.

Create containers and organizational units for ADAM

Objects stored in ADAM can be logically grouped into containers and organizational units (OU) just as they would in Active Directory.

Role required: admin

1. Right-click the root partition and select New > Object > organizationalUnit. You can also view the list of other objects that are available. This list varies based on the schema extensions installed when you imported the LDF files.

   Note: You can also view the list of other objects that are available. This list varies based on the schema extensions installed when you imported the LDF files.

2. When prompted for a value, enter the name of OU, for example Users.
3. The next screen displays a More Attributes button; use this to assign values to additional attributes. For OUs and containers, no additional values are needed.
   After creating OUs, the new OUs are listed as a child of the root object.

Delegation with ADAM

Once the OU structure is created, define the permission delegations to properly secure the objects to limited users.

As with Active Directory, there are two general ways to grant permissions:

- Add users to a group that already has the appropriate permissions assigned.
- Define new permissions on the ADAM objects.

For this task, we discuss object level permissions. Refer to the Group Administration section for information on group memberships.

Since we don’t have a Users and Computers console for ADAM, all object level permissions are defined using the Active Directory utility DSACLS.exe. This file is found in the ADAM program directory. When running ADAM utilities it is best to launch the ADAM Tools Command Prompt. This ensures the proper versions of the tools. DSALCS is used to view and set object access rights.

Example: "dsacls \localhost:50010\dc=myCompany,dc=adam" displays the permissions assigned to the root of partition dc=myCompany,dc=adam running on the localhost, port 50010. DSACLS is a complex tool used to create complex delegation. Run "DSACLS /?" for usage notes.

Populating ADAM Objects

ADAM Objects include User Objects, UserProxy Object, and Group Objects.
User Objects

Users can be created using the ADAM ADSI Edit console just as we did for OU creation. Users can also be administered using AD command line tools, which is beyond the scope of this document. The only mandatory attribute for new user objects is the cn, which is a short name or the user’s full name. There are also a wide range of optional attributes similar to Active Directory user attributes. You can access the full list of attributes by selecting properties from the user object.

UserProxy Objects

For ServiceNow LDAP integration we recommend you use UserProxy objects in ADAM which creates a proxy account that links to the related AD user account. This allows you to have ADAM authenticate logon credentials using AD usernames and passwords from the domain without ServiceNow directly connecting to the Domain Controller. UserProxy objects are very similar to AD and ADAM User objects except that do not store passwords and has an objectSID attribute that contains the SID from the linked AD User object. This is how the proxy works. UserProxy objects are created using the ADSIEdit console or command line tools, but this can be tedious. It is recommended that you use an automated process as defined below.

Group Objects

Groups are created using the ADSIEdit console and AD command-line tools. Group concepts are similar to AD and are used to integrate groups and members to ServiceNow. The biggest difference is ADAM groups can contain members from ADAM or from trusted AD Domains.

Automating ADAM Object Creation

If you are interested in synchronizing Active Directory accounts to ADAM, we recommend you use Microsoft ADAMSync tool. This is the most common use of ADAM for ServiceNow LDAP integration.

About Permission Delegation

ADAM contains some built-in groups with default permissions. These groups are found in the container cn=roles,dc=myCompany,dc=adam. These are similar to domain level groups and have rights to objects in the current partition. Similar to AD Forests you can also set a higher level of permissions using the default groups in cn=roles,cn=configuration,dc=myCompany,dc=adam. You must connect to the configuration partition in ADSIEdit. The Administrators group by default includes the account specified during the setup. This member is not always visible since it’s inherited through the configuration groups. Administrators have full control of all partition objects. The Readers group does not contain any members by default and has read access to all objects in the partition. The Users group is a dynamic group just as it is in Active Directory. Transitivity it includes all ADAM users created in the partition.

Testing and troubleshooting ADAM setup

The primary tool used for testing is LDP. This allows you to fully test user authentication.

Most of the object management can be completed using the ADAM ADSI Edit console which will provide access to the entire collection of objects and attributes. The highest level of control and troubleshooting ADAM services is using the Windows service created during the instance setup. The service name will vary and depends on the name of the instance created. This service must be running in order for the ADAM service to run. If you are experiencing connection problems, you should review the network configurations to ensure you have the appropriate network access to connect to the server and ADAM port. For each
ADAM instance installed, a Windows Event Log is created. This is also a great tool for troubleshooting ADAM services.

The Windows Security Event Log is also helpful when troubleshooting userProxy authentications. All userProxy logon attempts are logged in the Security Log and reference the remote client device address, the distinguished name of the user trying to log on, and the result or status code.

**Backup and recovery with ADAM**

All ADAM data can be backed up using standard file system backup methods.

We recommend following Microsoft procedures for restoring (recovery of) an ADAM instance.

**Redundancy**

ADAM has built-in replication utilities based on the same technology as AD. A full read and write replica of an ADAM partition can exist on the same or different computer. You can use this replica in a variety of ways to provide a fault-tolerant LDAP integration with the instance. One option is to expose both partitions to the instance through the firewall and define both servers in the LDAP Properties server field.

**Use LDAPS with ADAM**

The default configuration for userProxy object authentication is to enforce LDAPS (secure LDAP) communications. LDAPS requires SSL certificates to secure the network traffic.

To remove this requirement make the following change using the ADSIEdit console connected to the configuration partition.

```
Object: CN=Directory Service, CN=Windows NT, CN=Services, CN=Configuration
Attribute: msDS-Other-Settings
Value: change RequiresSecureProxyBind from 1 (enforced) to 0 (disabled)
```

Restart the ADAM service to use the new setting.

To support secure binds and encrypt the user and password information being transmitted, a SSL certificate must be installed on the server and any LDAP client. Since there is limited and controlled uses to the ADAM service, it is feasible to use a self-signed certificate which would meet the needs without incurring certificate costs or building a Certificate Authority (CA) infrastructure. If you already have a CA, you can issue a certificate. Otherwise, create a self-signed certificate.

**Creating a Self-Signed Certificate**

To use the selfssl utility, Internet Information Services (IIS) must be installed. This service can be removed after you generate the certificate. You can get the selfssl.exe utility from the IIS Resource Kit. If IIS is already installed, create a new website so that the current sites will not be impacted during the certificate generation. Selfssl needs to temporarily attach the new self-issued certificate to a valid web site. Selfssl is a command-line tool and has the following common parameters.

**Table 444: Selfssl Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/T</td>
<td>Adds the cert to ‘Trusted Certificates’ on the local machine</td>
</tr>
<tr>
<td>/N:cn</td>
<td>Set the common name of the certificate. This must match the fully qualified domain name of the server running the web service using the certificate</td>
</tr>
</tbody>
</table>
The common name attribute should match the external name or address that the instance will use to connect to your ADAM computer. You will need to get the IIS Website site id unless you are using the default website which is 1 and does not need to be defined in the selfssl command. A sample command to generate a certificate for myCompany would be:

```
selfssl /N:CN=myCompany.externaldomain.com /K:1024 /V:3650 /S:12345 /P:50001 /T
```

This statement creates a certificate that is valid for 10 years. Set the value to any duration, but be aware the new certificate must be generated and submitted to the instance before the old one expires. We recommend making a note of the expiration date on the certificate.

Once the certificate is generated you can remove it from the website, or delete the entire web site if you created a temporary site.

### Assigning the Certificate to ADAM

To support secure binds and encrypt the user and password information being transmitted, a SSL certificate must be installed on the server and any LDAP client.

Role required: admin

Since there is limited and controlled uses to the ADAM service, it is feasible to use a self-signed certificate which would meet the needs without incurring certificate costs or building a Certificate Authority (CA) infrastructure.

To assign the certificate to ADAM.

1. Open the Certificates MMC console. Create two console connections, one for Local Computer Certificates, and the other for Local Computer Services Certificates on the new ADAM service. The new certificate can be found under Certificates (Local Computer)\Personal Certificates.
2. Copy the certificate to the container for the ADAM service, Certificates – Service (ADAM Service Name)\ADAM_ADAM Service Name\Trusted Root Certificates\Certificates. Also copy the certificate to Certificates – Service (ADAM Service Name)\ADAM_ADAM Service Name\Personal\Certificates.
3. Open the details tab on the certificate you copied. Note the Valid from date stamp. Now assign read access to the certificate key file. Go to C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA\MachineKeys and identify the certificate with the matching time stamp. Assign Read & Execute rights to the service account running ADAM. By default this is Network Service.
4. Restart the ADAM service to activate the new certificate.

### Exporting the Public Key Certificate

LDAPS clients, including the instance need the public key certificate in order to make a secure connection to ADAM.

Role required: admin
From the server certificate consoles you used above, export a public key to be used by the clients.

1. Select the certificate, right-click, select **all tasks/export**. Do not export the private key. Select the default DER encoded binary X.509 format and specify the export file name.

2. Install the public certificate on the LDAP clients that connect to the server using LDAPS. When prompted, add the certificate to the **Trusted Root Certificate Authorities store**.

Test communication between LDAP and MID Servers
The instance tests the connection automatically every time a user opens the LDAP Server form.

Role required: admin

The instance supports an LDAP connection timeout of 29 seconds or less. Error messages appear on the form if there are any issues connecting to the LDAP server.

Employees can also verify connectivity between the instance and the LDAP server. Contact Technical Support for assistance verifying LDAP connectivity.

**ServiceNow ITSA Suite Access Account**
ServiceNow requires a user account to read the ADAM object information that is imported into the application instance.

Create the account by using one of the following methods:

- Create a local ADAM user account and assign it a password and assign permissions.
- Assign permission to a Windows domain account on the ADAM partition.
- Use a **userProxy** account.

When using ADAM as an LDAP source, you must specify the fully qualified distinguished name (FQDN) of the ADAM account in the instance’s LDAP server’s **Login distinguished name** field.

**Testing LDAPS Connections**
There are two console connections, one for Local Computer Certificates, and the other for Local Computer Services Certificates on the new ADAM service.

Role required: admin

1. Run **LDP.exe** from the ADAM install folder `c:\windows\adam`. Verify that the ADAM version is selected because this is not the standard Windows LDP client.

2. Open a new connection using the **Connection/Connect** menu. The server name must match the CN assigned to the certificate.

3. Enter the **LDAPS port** and select the **SSL** checkbox.
The results of a successful connection are some general server information and no errors.

4. Bind (log in) to the service. To replicate typical LDAP client connections select the Simple bind option. Enter a valid ADAM user or **userProxy** distinguished name in the user field and the associated password.
If you see a return message stating ‘Authenticated as:…..’ then you have successfully connected using LDAPS.

**Use ADAMSync to populate ADAM**
Administrators use MS ADAMSync to populate LDAP directories that use MS ADAM.

**Note:**
This document assumes you have at least a basic level of understanding with Microsoft Windows Server, Active Directory, and ADAM and that you already have a functional **ADAM** instance with a partition.
These are sample procedures. Due to the complexity and the fact that it is running in your environment, we cannot offer direct support. We recommend you work with Microsoft or a Microsoft consultant if you run into any trouble.

Once ADAM has been installed and the first partition has been created, you can populate it with objects. The following options are available:

- Manual object creation using GUI or scripts. This option is inefficient and slow.
- Integrate with Active Directory using Microsoft Integration Information Server. This option ultimately provides the most flexibility and functionality but does require some advanced configurations. There is a free version of MIIS available that is compatible with Active Directory, ADAM, and Microsoft Global Address Lists from Exchange. Unless you already have experience with MIIS we advise that you don’t attempt to implement a new environment for LDAP integration only.
- Use ADAMSync, a synchronization tool that Microsoft provides with ADAM. This is the option that is explained here.

Define ADAM user accounts

Define the following user accounts in ADAM. One is used for the instance to connect with and the other for ADAMSync.

Role required: admin

These accounts can be local ADAM User objects, UserProxy objects, or a Windows account from a trusted domain.

User Account

This account requires read-only access to the directory structure you are importing to your instance. The best way to accomplish this is to add the account to the member attribute on the Readers group found in cn=roles,dc=myCompany,dc=adam.

New ADAM User accounts are disabled by default. You will need to enable the new accounts and set a password.

1. Enable users by changing the attribute msDS-UserAccountDisabled to FALSE.
2. Right-click the user object and reset the password.
3. Test the new accounts using LDP as defined in ADAM to make sure they can connect. Use the LDAP > View/Tree option, leaving the Base DN blank to make sure you can view the objects in the directory using the new accounts. The Configuration, Schema, and the domain partition should be visible in the left pane. Traverse the domain partition. If you are using a new local ADAM account, it will show ‘No Children’ which means you don’t have read access to the objects. Verify the Setup group memberships and re-test.

ADAMSync User Account

ADAMSync uses this account to manage objects in the ADAM partition. This account requires admin level rights since it will create, update, and delete ADAM objects.

ADAMSync AD Account

ADAMSync uses this account to read the AD objects that will be synchronized to ADAM.

Set up ADAMSync

ADAMSync is included with Windows Server 2003 R2. Download and install ADAMSync if you are using a different OS.

Extending the schema

The ADAM schema needs to be extended to support ADAMSync.
1. Run the following command from c:\windows\adam to import the ADAMSnc schema extensions. You may have to change the server:port and add credentials if the current user doesn't have access. See the AdamSyncMetadata.ldf file for details.

```
ldifde -i -f MS-AdamSyncMetadata.LDF -s localhost:50000 -j . -c "cn=Configuration,dc=X" #configurationNamingContext
```

2. Do the same with MS-AdamSchemaW2k3.ldf to support Windows 2003 attributes.

```
ldifde -i -u -f MS-AdamSchemaW2K3.LDF -s localhost:50000 -j . -c "cn=Configuration,dc=X" #configurationNamingContext
```

**Recommended schema changes**

Here are some additional schema changes we recommend.

1. Open a new MMC console and add the ADAM Schema Snap-in.
2. Connect to the ADAM instance.
3. Expand the Classes folder and locate the userProxy class, open **Properties**.
4. Verify the following optional attributes on the Attributes tab, add any that do not already exist.
   - company
   - department
   - givenName
   - mail
   - physicalDeliveryOfficeName
   - sAMAccountName
   - sn
   - telephoneNumber
   - title
   - userAccountControl
   - userPrincipalName

5. Restart the ADAM Service to enable the new settings.

**Install the ADAM configuration file**

You can install the ADAM configuration file through the Windows command line.

Role required: admin

1. Install the configuration file.

```
C:\WINDOWS\adam>adamsync /install localhost:50000 MS-AdamSyncConfSNC.XML
```

2. Run the synchronization file. This will log to the console and may run for a long time.

```
C:\WINDOWS\adam>adamsync /sync localhost:50000 "ou=users,dc=service-now,dc=adam" /log -
```

3. Review the results by using the ADSIEdit console. You should see the new objects and attributes that were created by ADAMSnc.
4. Run ldap to test the UserProxy authentication.
Automating the sync process

Setup the sync process as a Windows Scheduled Task. You must either provide the credentials in the config file, command line, or run the Scheduled Task with an account that has access.

Special notes

- You can create multiple configuration files and scheduled jobs to sync ADAM from multiple sources.

  This example imports the sAMAccountName attribute which can be used as the application logon. If you are going to sync source you need to make sure you have a unique attribute value that can be used for the logon credentials. sAMAccountName is guaranteed to be unique within a domain, but not across multiple domains.

- If you are using Microsoft Exchange, we recommend excluding cn=SystemMailbox* objects as part of the object-filter configuration.

Example ADAM configuration files

All of the configurations for ADAMSsync are stored in xml files.

**Default configuration file with comments**

There is a default configuration file called MS-AdamSyncConf.xml included with the ADAMSsync install. Make a copy of this file so you have a base example to refer to in the future. This example is the default configuration file with comments added.

```xml
<?xml version="1.0"?>
<doc>
  <configuration>
    <!-- Sync File Description -->
    <description>MyCompany ADAMSync Configuration</description>
    <security-mode>object</security-mode>;
    <!-- source-ad-name = fqdn of the domain controller -->;
    <source-ad-name>;fully.qualified.domain.name.of.domain.controller</source-ad-name>;
    <!-- source-ad-partition = root AD domain partition -->;
    <source-ad-partition>;dc=myCompany,dc=com</source-ad-partition>;
    <!-- source-ad-account = use this to specify an account to connect to AD --
    >; <source-ad-account>;</source-ad-account>;
    <account-domain>;</account-domain>;
    <!-- target-dn = target ADAM OU -->;
    <target-dn>;ou=servicenow users,dc=myCompany,dc=adam</target-dn>;
    <query>;
    <!-- base-dn = should be the root AD partition if you want all users -->;
    <base-dn>;dc=myCompany,dc=com</base-dn>;
    <!-- object-filter = standard ldap query format, this will grab all users --
    >; <object-filter>;(&objectCategory=person)</object-filter>;
    <attributes>;
    <!-- include=userproxy requires objectSID to link back to the AD account --
    >; <include>;objectSID</include>;
    <include>;givenName</include>;
    <include>;sn</include>;
    <include>;description</include>;
    <include>;title</include>;
    <include>;company</include>;
```
LDAP filters configuration file

You can provide any level of filtering in the object-filter value in the configuration file. Use standard LDAP query syntax with the following xml escape characters in place of the standard operators.

- **AND** = "&" replace with &\#38;
- **OR** = "|" (vertical line) replace with &\#124;
- **NOT** = "!" replace with &\#33;

Reference configuration file

Here's an actual configuration file that can be referenced as a sample.

```xml
<?xml version="1.0"?>;
<doc>;
<configuration>;
<description>;SNCTest ADAMS Syn Configuration</description>;
<security-mode>;object</security-mode>;
<source-ad-name>;domaincontroller.service-now.com</source-ad-name>;
<source-ad-partition>;dc=service-now,dc=com</source-ad-partition>;
<source-ad-account>;\</source-ad-account>;
<account-domain>;</account-domain>;
</configuration>;
</doc>
```
Configure Microsoft Active Directory for secure LDAPS communication

Use certificate pairs to enable Microsoft Active Directory (AD) LDAPS communications.

**Note:** These procedures were designed and tested using Windows 2003 R2 Standard Edition and work with all versions of Windows 2003.

Secure LDAP (LDAPS) communication is similar to SSL (HTTPS) communication in that both encrypt the data between servers and clients. To accomplish this, the server and clients share common information by using certificate pairs. The server holds the private key certificate and the clients hold the public key certificate. These certificates are required to enable Microsoft Active Directory (AD) LDAPS communications.
To configure LDAPS for Active Directory you must:

- Ensure that the Active Directory domain is set up and that the instance is able to connect to the Active Directory server through the firewall.
- Verify that there is a Certificate Authority (CA) that can issue a certificate for the domain controller (DC). If you don't already have a CA infrastructure there are two options.
  - Setup a stand-alone CA to issue the certificate
  - Request a third party certificate
- If you already have a CA in place, you can generate a certificate from an internal CA.

All certificates have a defined expiration date which can be viewed in the certificate properties. If the certificate expires, all LDAPS traffic fails, and your users can no longer log into the instance. To resolve this, a new certificate must be issued and installed on your instance.

The default expiration for Microsoft CA certificates is one year. External CA certificates are usually purchased in one year increments. Note when your certificate expires, or use the application's Expiration Notification function (located in System LDAP > Certificates). Ensure that you have a new certificate ready before the old one expires. This gives you time to install and test the new certificate before the old one expires.

**Set up a stand-alone certificate authority**

The first step to configure Microsoft Active Directory for SSL access is to set up a stand-alone Certificate Authority (CA).

Role required: admin

Do not worry about addition resource utilization because both of the required services (IIS & CA) can be disabled after issuing the certificate(s).

1. Install Internet Information Server (IIS).
2. Install Certificate Authority Services in stand-alone mode.
3. Verify the Certificate Services web application is installed and active.

Using the IIS Manager console, expand the local computer and select Web Sites. The state of Default Web Site should be Running. You should also see a CertSrv application listed under the Default Web Site. If the site is not running or the application is missing, you must resolve the issue before you proceed.

**Generate a certificate from an internal certificate authority**

When you configure Microsoft Active Directory for SSL access, you must generate an internal certificate and request the external certificate.

Role required: admin

These steps apply to Microsoft CA services. If you have a different internal CA platform, see your local CA administrator for assistance.

1. From the domain controller (DC) you want to create a certificate for, browse to http://localhost/certsrv or specify the CA server name if it is on a remote server.
2. From the Welcome page, click Request a certificate and select advanced certificate request.
3. On the Advanced Certificate Request page, select Create and submit a request to this CA.
4. Complete the Advanced Certificate Request as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The fully qualified domain name (FQDN) of the DC that is requesting the certificate.</td>
</tr>
<tr>
<td>Field</td>
<td>Entry</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>E-Mail</td>
<td>The email address of the person responsible for the certificate.</td>
</tr>
<tr>
<td>Company</td>
<td>Your company name.</td>
</tr>
<tr>
<td>Key Options settings</td>
<td></td>
</tr>
<tr>
<td>Create new key set</td>
<td>Select it.</td>
</tr>
<tr>
<td>CSR</td>
<td>Microsoft RSA SChannel Cryptographic Provider.</td>
</tr>
<tr>
<td>Key Usage</td>
<td>Exchange.</td>
</tr>
<tr>
<td>Key Size</td>
<td>1024 is recommended. The instance supports up to 2048.</td>
</tr>
<tr>
<td>Automatic key container name</td>
<td>Select it.</td>
</tr>
<tr>
<td>Store certificate in the local computer certificate store</td>
<td>Select it.</td>
</tr>
</tbody>
</table>

5. Click **Submit**.
   You are directed to a page that provides your **Request ID**, make note of this ID.

6. To process the pending request, complete the following:
   a) Open the Certificate Authority management console.
   b) Expand the server node and select **Pending Requests**.
   c) Locate the Request ID for the request you just submitted, right-click, and select All Tasks/Issue to approve the request and issue the certificate.

7. To retrieve the issued certificate, complete the following:
   a) From the DC you made the request from, browse to `http://localhost/certsrv`, or specify the CA server name if it is on a remote server.
   b) Select **View the status of a pending certificate request**.
   c) Select the link to the new certificate.
   d) Select the link to **Install this certificate**.

You need to request a third party certificate. Certificates from external CAs can be purchased for as little as $30 per year. For detailed procedures on requesting a certificate from an external CA, see Microsoft article [321051](http://support.microsoft.com/articles/321051).

After it is received, installed, and tested, follow the export procedure.

**Test the LDAPS Connectivity Locally**

When you configure Microsoft Active Directory for SSL access, you must test the DLAPS connectivity after installing the internal and third party certificates.

Role required: admin

1. Ensure that Windows Support Tools are installed on the domain controller (DC).
   The Support Tools setup (suptools.msi) can be found in the `\Support\Tools` directory on your Windows Server CD.

2. Select **Start > All Programs > Windows Support Tools > Command Prompt**. On the command line, type `ldp` to start the tool.

3. From the `ldp` window, select **Connection > Connect** and supply the local FQDN and port number (636). Also select the SSL.
If successful, a window displays and lists information related to the Active Directory SSL connection. If the connection is unsuccessful, try restarting your system and repeat this procedure.

*Export the public key certificate to trust the LDAP certificate*
When you configure Microsoft Active Directory for SSL access, you must export the public key certificate and import it into the application.

Role required: admin

If your Certificate Authority is not a trusted third party vendor, you must export the certificate for the issuing CA so we can trust it, and, by association, trust the LDAP server certificate. For MS Certificate Services users, you can view the certificate path by viewing the certificate in the console used to export; select the **Certificate Path** tab. You must export all certificates in the chain. You can find the CA certificate in the same folder as the LDAP certificate by looking for the name in the Certificate Path. Submit all certificates for importing to your instance.

1. From a current or new MMC console, add the Certificate (Local Computer) snap-in.
2. Open the **Personal/Certificates** folder.
3. Locate the new certificate. The **Issued To** column shows the FQDN of the domain controller.
4. Right-click the certificate and select **All Tasks/Export**.
5. Export to DER or Base-64 format. Name the file using the format: `MyCompany.cer`.
   - This is the public key certificate that needs to be used on the instance to communicate securely with your domain controller.
6. Test LDAPS locally before you submit the certificate to the instance.

After completing this procedure, import the public key certificate into the application.

See [*Upload the LDAP X.509 SSL certificate*](#) on page 1823 to upload the certificate into the application.

*LDAP global catalog usage*

A DC can be granted the Global Catalog (GC) role. Global Catalog (GC) role is an LDAP-compliant directory consisting of a partial representation of every object from every domain within a forest.

Administrators configure Active Directory to host Lightweight Directory Access Protocol (LDAP) directory information using one of the following hosting methods.

- The common method of hosting LDAP directory information is to use the default LDAP or LDAPS (secure LDAP) on ports 389 or 636. These standard LDAP ports always exist on a Domain Controller (DC) and are rarely changed. Accessing this directory partition provides access to all of the objects within the domain that is hosted on the DC. There is no way to access objects from other domains using this method.
- A DC can also be granted the Global Catalog (GC) role. Global Catalog (GC) role is an LDAP-compliant directory consisting of a partial representation of every object from every domain within the forest. This LDAP directory can be accessed on port 3268, with LDAPS on port 3269. LDAPS and the default LDAP ports’ certificate requirements are the same.

*Global Catalog LDAP dependencies*

- The domain controller that your instance connects to must have the Global Catalog role enabled.
- Firewall rules must allow inbound traffic to the domain controller on port 3268 (LDAP) or 3269 (LDAPS).
Special notes

- Not all attributes are replicated to the GC partition. Common attributes such as first name, last name, email, phone number, description, and address are included. Additional attributes can be added to the GC but should be limited to minimize the impact to forest replication traffic.
- Standard LDAP integrations usually use sAMAccountName as the instance's UserID and as the coalesce key in the LDAP import map since this is guaranteed to be unique within a domain. This attribute is no longer unique when viewing an entire forest of domains. A new unique attribute needs to be identified and as the UserID and the coalesce key. These do not need to be the same attribute and may vary based on your forest design. Consult your Active Directory administrator. Typically, the userPrincipalName is a unique attribute across domains but this may not be a user-friendly name to login with, but it could be used for the unique identifier on imports. A common attribute that is used for the UserID is email address. These decisions impact the LDAP Properties and LDAP Mapping.
- The value used for the coalesce key on the LDAP import map must be unique and exist on every object being imported. If it is not unique or does not exist, incorrect records are updated with changes.
- If you already have an LDAP integration and wish to change it to a GC, change the import coalesce key. The new key values must be imported before you can change the coalesce key.
- If you make any changes to your LDAP integration that break your integration, your first step should be to revert those changes. After that, contact Customer Support with complete information about what you're attempting.

OpenLDAP minor schema modification

In OpenLDAP 2.3 systems that use the back-bdb (Berkley backend), administrators make a minor modification to their schema to facilitate the integration.

Caution: The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

In OpenLDAP 2.3, back-bdb has limited support for inequality indexing (ordering). It is implemented only for generalizedTime and ChangeSequenceNumber syntax. It cannot be supported on syntax that support substrings. Search filters containing inequalities are processed using the presence index.

We recommend creating a custom attribute for this purpose, instead of changing what is already indexed or present in the schema (for example, servnowid).

Modify the OpenLDAP schema

These steps detail a schema modification to OpenLDAP 2.3 provided by one of our customers that helped them integrate with their instance.

Role required: admin

Caution: The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

To modify the OpenLDAP schema for integration with the instance:

1. Create a custom attribute.

   ```
   attribute { 1.3.6.1.4.1.3403000.2.1.8
                  NAME 'servnowid'
                  ORDERING caseIgnoreOrderingMatch
   ```
2. Include the attribute in the selected objectclass OID.

   objectclass { 1.3.6.1.4.1.3403000.2.2.1
                 NAME 'BcfUserIdentifiers' SUP top AUXILIARY
                 MAY ( uniqid $ unixid $ servnowid )
   }

In OpenLDAP 2.3, you can dynamically change the server configurations, but you can only extend the schema. You cannot modify or delete the existing schema. Instead of creating another objectclass for this attribute in the dynamic configuration, use the static configuration file, slapd.conf.

3. In slapd.conf, include indexing for the new attribute in the bdb section of your main database backend.

   database bdb (configs here) ....
   index servnowid pres
   (other indexes here) .....  

4. As root, run slapindex to index this attribute to make it available in search filters. Make sure that the OpenLDAP daemon is not running or is in read-only mode before starting slapindex.

Record LDAP deletions

By default, the instance does not delete any entries after they disappear from LDAP.

Deleting an entry, also referred to as a record, also deletes the entire history and references to the deleted entry.

For example, configuration items (CIs), SLA agreements, software licenses, purchase orders, and service catalog entries all have a reference to Department, and if a department is deleted, then the integration clears all references to the department. Also, deleting a user results in losing all history of what that user did.

Decide whether to retain or manually delete records according to your organization's needs.

Multiple provider single sign-on

The multiple provider single sign-on (multi-SSO) feature allows organizations to use several SSO identity providers (IdPs) to manage authentication as well as retain local database (basic) authentication.

The integration supports any combination of local and external authentication methods on a single instance:
- SAML 2.0
- Digest Authentication
- LDAP
- Local database authentication

For example, a globally dispersed corporation might require one SSO provider for their employees, a different one for their vendors, and local database authentication for their administrators. Alternatively, a company might implement SAML 2.0 and a digest token authentication solutions on the same instance.

Changes to SAML 2.0 and digest token configuration

Multiple provider single sign-on allows administrators to configure SAML 2.0 Update 1 and digest token as authentication methods. Multiple provider single sign-on should be activated before you configure your
SAML 2.0 Update 1 and digest token properties. After you activate multi-provider SSO, you must then set it up. After setting up multi-provider SSO, you can create or update the SAML 2.0 Update 1 and digest token configurations. You can use either or both authentication solutions with multi-provider SSO.

**Note:** The Integration - Multiple Provider Single Sign-On Installer plugin removes the SAML application from the navigator. The necessary SAML settings are migrated to the Multi-Provider SSO application into the SAML2 Migrated table. You can still modify items like the x509 certificate, IdP details, and so on through the Multi-Provider SSO application.

---

**Use E-Signature with Multi-Provider SSO**

When approval with e-signature is active, approving a request, like a change or a service catalog order, usually requires the approver to enter their credentials. After you configure multi-provider SSO, approvers enter their IdP login credentials instead.

When Multi-Provider SSO is enabled, make sure to configure the Identity Provider form and add the **Assertion Consumer URL for eSignature authentication** field. In most cases, this URL will be: https://YOURINSTANCE.service-now.com/consumer.do. However, if you employ a customized method of handling the SAML authentication for eSignature, you can set up your own consumer URL.

If you are only using SAML 2.0 Update 1 and not using Multi-Provider Single Sign-on, configure the assertion consumer URL with **E-signature SAML properties**.

**Local database authentication with Multi-Provider SSO**

If you still want to use local database authentication when Multi-SSO is active, you must set the glide.authentication.external.disable_local_login property to `false`. See **Redirection Properties**.

**Installed with multi-provider SSO**

The following items are installed with the Integration - Multiple Provider Single Sign-On Installer plugin.

**Properties**

Multi-Provider SSO adds the following system properties.

### Table 446: Multi-provider SSO properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.authenticate.multisso.debug</td>
<td>Enables (true) or disables (false) debug logging for the multi-provider SSO integration.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td>glide.authenticate.multisso.enabled</td>
<td>Enables (true) or disables (false) multi-provider SSO.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
</tbody>
</table>
Tables

Multi-Provider SSO adds the following tables.

Table 447: Multi-provider SSO tables

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSO Properties [sso_properties]</td>
<td>Stores data for each IdP, schema, common SSO data, and so on.</td>
</tr>
<tr>
<td>SAML 2 Update 1 Properties</td>
<td>Stores data for SAML 2.0 Update 1 configurations such as SAML certificates.</td>
</tr>
<tr>
<td>Digest Properties [digest_properties]</td>
<td>Stores data for digest token authentication configurations.</td>
</tr>
<tr>
<td>SSO Federation [sso_federation]</td>
<td>Stores data for each SSO federation.</td>
</tr>
</tbody>
</table>

Scripts

Multi-Provider SSO adds the following scripts.

Table 448: Multi-provider SSO scripts

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiSSO</td>
<td>Allows a customer to have an SSO type defined on a company basis.</td>
</tr>
<tr>
<td>MultiSSOLogin</td>
<td>Allows each domain to have their own login script.</td>
</tr>
<tr>
<td>MultiSSOLogout</td>
<td>Allows each domain to have their own logout script.</td>
</tr>
<tr>
<td>MultiSSO_Abstract_Core</td>
<td>Provides a base class for all multi-provider SSO classes.</td>
</tr>
<tr>
<td>MultiSSO_ClientHelper</td>
<td>Provides a client callable utility functions for multi-provider SSO.</td>
</tr>
<tr>
<td>MultiSSO_DigestedToken</td>
<td>Provides a base system logic for digested token authentication.</td>
</tr>
<tr>
<td>MultiSSO_SAML2_Update1</td>
<td>Provides logic to process SAML 2.0 Update 1 authentication for a multi-tenant single sign-on.</td>
</tr>
</tbody>
</table>

Activate multiple provider single sign-on

This integration requires the Integration - Multiple Provider Single Sign-On Installer plugin.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.

If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.

Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click Activate.

Set up multi-provider SSO

You must perform several steps to set up Multi-SSO, including configuring properties, creating identity providers (IdPs), and configuring users to use Multi-SSO.

Create a SAML 2.0 update 1 SSO configuration for Multi-SSO

You can create and update SAML 2.0 Update 1 SSO configurations from the multiple provider single sign-on feature.

Role required: admin

1. Navigate to Multi-Provider SSO > Identity Providers.
2. To update a configuration, click an SSO configuration record.
3. To create a new configuration, click New.
4. Click SAML2 Update 1.
5. Complete the form, using the fields from the table and submit the record.

For more information on multi-provider single-sign on, see Multi-Provider Single Sign-On.

Table 449: Multi-provider single sign-on fields

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name for the SSO property record.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to set the SAML configuration to active.</td>
</tr>
<tr>
<td>User Field</td>
<td>Enter the field on the User table that contains the value the IdP needs to identify the user.</td>
</tr>
<tr>
<td>Identity Provider URL</td>
<td>Enter the URL to your IdP.</td>
</tr>
<tr>
<td>Identity Provider's AuthnRequest</td>
<td>Enter the URL to the HTTP-Redirect binding obtained from the SingleSignOnService element. Add the value to the glide.security.url.whitelist property.</td>
</tr>
<tr>
<td>Identity Provider's SingleLogoutRequest</td>
<td>Enter the URL obtained from the SingleLogoutService element.</td>
</tr>
<tr>
<td>Failed Requirement Redirect</td>
<td>Enter the URL for redirecting failed authentication requests. Typically, the URL endpoint is an error page or logout page.</td>
</tr>
<tr>
<td>ServiceNow Homepage</td>
<td>Enter the URL, including login page, of the instance for which the IdP authenticates. For example: <a href="https://yourinstance.service-now.com/navpage.do">https://yourinstance.service-now.com/navpage.do</a></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Entity ID/Issuer</td>
<td>Enter the base URL, excluding login page, of the instance for which the IdP authenticates. For example: <code>&lt;nowiki&gt;https://yourinstance.service-now.com&lt;/nowiki&gt;</code></td>
</tr>
<tr>
<td>Protocol Binding for the IDP’s SingleLogoutRequest</td>
<td>Enter one of the supported values listed in the Binding attribute from the SingleLogoutService element.</td>
</tr>
<tr>
<td>NameID Policy</td>
<td>Enter the value of the NameIDFormat element the integration uses.</td>
</tr>
<tr>
<td>NameID Attribute</td>
<td>Leave this field blank unless you configure a new NameID policy. If you do configure a new policy, the system needs to know which field in the User table it should use to identify the user logging in by matching the NameID token. Enter the name of that User table field here.</td>
</tr>
<tr>
<td>Create AuthnContextClass</td>
<td>Select the check box to specify a particular context class such as Password Protected Transport. If the check box is cleared, the IdP selects the most appropriate context class.</td>
</tr>
<tr>
<td>AuthnContextClassRef Method</td>
<td>Enter the URN of the login mechanism you want the IdP to use to authenticate users.</td>
</tr>
<tr>
<td>External logout redirect</td>
<td>Enter the URL where the integration redirects users after they log out.</td>
</tr>
<tr>
<td>Signing/Encryption Key Alias</td>
<td>Enter the alias of the key entry stored in SAML 2.0 SP Keystore.</td>
</tr>
<tr>
<td>Signing Key Password</td>
<td>Enter the password of the key entry stored in SAML 2.0 SP Keystore.</td>
</tr>
<tr>
<td>Encrypt Assertion</td>
<td>Select the check box to encrypt the assertion in the SAML response. The metadata generated for the IdP embeds the x509 certificate, which the IdP uses to encrypt the assertion in the SAML response that it generates.</td>
</tr>
<tr>
<td>Clock Skew</td>
<td>Enter the number of seconds between the two attributes that make up the SAMLResponse nonce. A valid SAMLResponse must fall between the notBefore and notOnOrAfter date-time values. See Sample SAML 2 Response with the SubjectConfirmation and SubjectConfirmationData Elements and Sample SAML 2 Response with the AudienceRestrictions and Audience Elements for a sample SAMLResponse message.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Force AuthnRequest</td>
<td>Select the check box to force AuthnRequests to occur.</td>
</tr>
<tr>
<td>Is Passive AuthnRequest</td>
<td>Select the check box if the AuthnRequest is passive.</td>
</tr>
<tr>
<td>Sign AuthnRequest</td>
<td>Select the check box to enable the IdP's single-sign on service to receive a signed AuthnRequest.</td>
</tr>
<tr>
<td>Signing Signature Algorithm</td>
<td>Enter the URL that points to the SAML 2.0 Identity Provider AuthnRequest Consumer for eSignature Authentication.</td>
</tr>
<tr>
<td>Auto Provisioning User</td>
<td>Enable automatic user provisioning, which creates a user in the instance's User table when the user exists on the IdP but does not exist in the User table.</td>
</tr>
<tr>
<td>Update User Record Upon Each Login</td>
<td>Updates user information in the instance's User table with the information in the IdP each time the user logs in using SAML.</td>
</tr>
<tr>
<td>Related list</td>
<td></td>
</tr>
<tr>
<td>X.509 Certificates</td>
<td>The IdP certificates. You can add as many certificates as necessary after you install them. When there are multiple certificates, the system uses the first active certificate that is found.</td>
</tr>
</tbody>
</table>

Configure multi-provider SSO properties
Configuring Multi-Provider SSO Properties.

Role required: admin

1. Navigate to Multi-Provider SSO > Properties.
2. Select the Enable Multi-Provider SSO check box. This adds the link Use external login to the login page.
3. To enable the debug messages to appear at the bottom of the content frame, select the Enable debug logging for the Multi-Provider SSO integration check box.
   If enabled, the debug logging feature will slow down performance and use up disk space in order to generate the logs.
4. In the property The field on the user table that identifies a user accessing the User identification login page, enter the field on the User table that contains the value the IdP needs to identify the user. The default value is user_name.
5. Click Save.
6. Instruct your users to click the Use external login link when they log in to the instance.
7. Open the System Properties [sys_properties] table and add the following property:
   glide.security.url.whitelist
   Add the Identity Provider’s AuthnRequest URLs to the whitelist, separated by commas.
Create and update identity providers
After you have configured the multi-provider SSO properties, you can update or create new SAML 2.0 Update 1 or digest token identity providers.

Role required: admin

1. Navigate to Multi-Provider SSO > Identity Providers.

2. To edit an identity provider record, click the record.
   • For digest token configurations, manually update the properties.
   • For SAML2 Update 1 configurations, automatically update the identity provider metadata with the Import Identity Provider Metadata related link or update the properties manually.

3. To create a new identity provider, click New.
   • For digest token configurations: Click Digest SSO and enter the digest properties for multi-provider single sign-on.
   • For SAML2 Update 1 configurations: Click SAML2 Update 1 and import the identity provider metadata from a URL, as XML, or manually enter the identity provider information.

4. To make the IdP the failover IdP that is used when the default IdP is not available, select the Default check box.

   If you have SAML 2 Update 1 active and you upgrade to the Fuji release, the SAML 2 Update 1 IdP is selected as the default failover. No default failover IdP is selected for new instances or if you are upgrading from a release on which SAML 2 Update 1 is not active.

**Note:** The metadata import process automatically creates a certificate record for the identity provider. Navigate to the x509 Certificate module to see the certificate.

**Note:** Certificates for single-sign on should always be in PEM format to work with SAML certificates.
5. If E-Signature is active, configure the Identity Provider form and add the **Assertion Consumer URL for eSignature authentication** field.

   In most cases, this URL is: `https://YOURINSTANCE.service-now.com/consumer.do`. However, if you employ a customized method of handling the SAML authentication for E-Signature, you can set up your own consumer URL. If you are only using SAML 2.0 Update 1 and not using Multi-Provider Single Sign-on, configure the assertion consumer URL with **E-signature SAML properties**.

Modify the primary and default IdP

You can set one IdP as the primary IdP to which new users are automatically redirected when they access the base instance URL. You can also set one IdP as the default IdP.

Role required: admin

The **Integration - Multiple Provider Single Sign-On Installer** plugin must be active.

1. To set one IdP as primary, create the following property and enter the IdP record's sys_id as the value:
   - **Name**: glide.authenticate.sso.redirect.idp
   - **Type**: String
   - **Value**: sys_id of the identity provider record

2. To set a default IdP, open the IdP record and select the **default** option.

   When this default is set to true, and a user with no SSO configuration selects **Use external login**, the user is redirected to the default IDP.

**Configure users for multi-provider SSO**

Administrators can configure Multi-Provider SSO for individual users or for all users who belong to a company. You cannot configure Multi-Provider SSO for groups.

Role required: admin

1. Navigate to **Multi-Provider SSO > Identity Providers**.
2. Right-click an identity provider record and select **Copy sys_id**.
3. Copy the data to your clipboard.
4. Navigate to a user record or a company record.
5. Configure the form and add the **SSO Source** field.
6. In the **SSO Source** field, enter one of the following:
   - **SAML users**: enter `sso:` followed by the sys_id of the identity provider's record.
   - **SSO Federation users**: enter `federation:` followed by the sys_id of the federation record.

7. Click **Update**.

**Test IdP connections**

Administrators can test the connection to an identity provider to validate settings before enabling external authentication.

Role required: admin

---

**Note**: The test connection user interface only works for identity providers that support displaying content in a frame. Some identity providers might require additional configuration to list your instance URL as an allowed origin in the X-Frame-Options response header. Contact your identity provider for more information.

---

1. Navigate to **Multi-Provider SSO > Identity Providers**.
2. Select an identity provider record.
3. Click Test Connection.
4. Enter login credentials for the identity provider and login.
5. Use the Testing SSO Logs section to see log messages.
6. Click Cancel when testing is complete.
Test SAML Connection (SSO: http://idp.devscn.com/openam)

SSO Login Test Results
✅ SAML Login response received
✅ SAML Assertion retrieved
❌ Unable to evaluate key against signature
  - Unable to evaluate key against signature
  - Ensure that the IDP's x509 certificate is present, valid and active.

SSO Logout Test Results
✅ SAML Logout response received
✅ SAML Logout Response 'InResponseTo' validated
✅ SAML Logout Response 'Status' validated

Testing SSO Logs
12/17/14 14:40:05 (324) Session InResponseTo: SNCeabe9f9bede11555f8b8f2992e974a7
12/17/14 14:40:05 (325) It is a logout response
12/17/14 14:40:05 (326) SAML2 LogoutResponse validated
12/17/14 14:40:05 (328) request type: LogoutResponse
12/17/14 14:40:05 (330) We will be redirecting user to the URL: /saml_test_conn_logout_completed.do?sysparm_noStack=true&sysparm_test_sso_id=71098dce1d741310b4c14e187e960108344
12/17/14 14:40:05 (334) userToLogin: logout_success

© 2017 ServiceNow. All rights reserved.
Geneva    ServiceNow    ServiceNow Platform

Common connection errors
The following table describes some of the common connection errors and their solutions.

Table 450: Troubleshooting IdP test connections

<table>
<thead>
<tr>
<th>Error messages</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Field validation failed. Invalid User Field</td>
<td>Verify the contents of the User table field you selected matches the SAML NameID token.</td>
</tr>
<tr>
<td>Assertion issuer is invalid.</td>
<td>Verify Identity Provider URL contains a valid URL to your IdP.</td>
</tr>
<tr>
<td>AudienceRestriction validation failed.</td>
<td>Verify the Audience URI contains a valid URL to your instance.</td>
</tr>
<tr>
<td>Cannot logout of IdP’s session.</td>
<td>Verify the SingleLogoutRequest URL contains a valid URL to your IdP’s logout service.</td>
</tr>
<tr>
<td>Signature did not validate against the credential’s key.</td>
<td>Verify the IdP has a valid certificate installed.</td>
</tr>
</tbody>
</table>

Troubleshoot script issues with SAML
You might encounter script issues if SAML is already active at the time you activate Multiple Single Sign-On and if you already customized the installation exits.

Role required: admin
1. Back up the modified installation exit SAML2SingleSignon_update1 and script include SAML2_update1.
2. Revert both the installation exit and script include to the version that is available with the baseline system.
3. Activate or upgrade the Integration - Multiple Provider Single Sign-On Installer plugin. The system upgrades SAML and all necessary files to SAML 2 Update 1.
4. Open the Multiple SSO properties page and select the Enable Multi-Provider SSO check box to enable it.
5. Put the SAML2SingleSignon_update1 installation exit changes into the baseline script include MultiSSO_SAML2_Update1 and the SAML2_update1 script include changes into the baseline SAML2_update1 script include.

Log in using Multi-SSO
The recommended and most efficient method for users to log in using multi-provider SSO is to use a specifically configured URL.

Role required: admin
After multi-provider SSO is configured, you can send a URL to your users with the correct IdP in the parameter string. For example:

```
/login_with_sso.do?glide_sso_id=<sys_id of the sso configuration>
```

After a user successfully logs in to the IdP page, a cookie containing the IdP sys_id is added to the browser. The next time the user attempts to log in, the system redirects the user to log in to the IdP server, which automatically logs in to the instance.

If a URL parameter is not set or the browser cache has been cleared, users can also do the following:
1. Click the Use external login link on the login page.
   The external login page appears. Users can click Use local login to return to the standard login page.
2. Enter the value for the specified field on the user table that you configured in Multi-Provider SSO properties.

The user is redirected to the IdP server, where they log in.

After users successfully log in to an IdP, they are automatically redirected to that IdP whenever they attempt to access the instance. To have a user access a different IdP, send the user a URL with the new IdP information in the parameter. The new IdP overwrites the old IdP in the cookie if the user successfully logs in. If the user does not log in successfully, the old IdP information is retained in the cookie.

Allow users to choose the identity provider for login

SSO federation support allows users to choose which IdP to log into.

Role required: admin

SSO federations aggregate metadata from multiple IdPs and service providers, including your instance. Federations then publish the metadata as an XML file, which includes information like IdP names and IdP certificates. Administrators can then instruct the instance to read the XML file and automatically populate the SSO Properties table with all the necessary IdP information.

1. Navigate to Multi-Provider SSO > SSO Federation.
2. Click New.
3. Fill in the fields, as appropriate (see table).
4. Click Submit.
5. After you configure a federation, enable the Refresh SSO Metadata scheduled job, and then configure the users who you want to access the federation IdPs. Use the sys_ID of the federation record you just created.

<table>
<thead>
<tr>
<th>Field</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Active</td>
<td>✔</td>
</tr>
<tr>
<td>Type</td>
<td>SAML</td>
</tr>
<tr>
<td>Discovery Service URL</td>
<td></td>
</tr>
<tr>
<td>Meta Data URL</td>
<td></td>
</tr>
<tr>
<td>x509 Certificate</td>
<td></td>
</tr>
</tbody>
</table>

The instance populates the SSO properties table with the IdP information. When users who are configured to use the federation log in, they are redirected to the discovery service URL you configured. Then they select the IdP and provide the necessary credentials. Alternatively, you can send users a URL with the IdP in the parameter.
Table 451: Allowing users to choose the identity provider for login

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for the federation.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to enable the instance to pull the XML file from the federation.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the type of authentication this federation supports.</td>
</tr>
<tr>
<td>Discovery Service URL</td>
<td>Enter the URL of the discovery service for this federation. This is the site where users are directed to select an IdP and log in.</td>
</tr>
<tr>
<td>Meta Data URL</td>
<td>Enter the URL of the XML file that holds the federation metadata.</td>
</tr>
<tr>
<td>x509 Certificate</td>
<td>Select the federation certificate.</td>
</tr>
<tr>
<td>Domain</td>
<td>Select the domain that the data will belong to.</td>
</tr>
</tbody>
</table>

**Note:** The InCommon federated identity management IdP is preconfigured.

*Use ESS pages with multi-provider SSO*

You can redirect ESS users to an employee self-service page by adding a system property.

Role required: admin

You can add a global property to the system properties table. It applies to all IdPs. You cannot set multiple values for different IdPs.

1. Add the following system property: `glide.entry.loggedin.page_ess`.
2. Set the value of this property to the ESS page name: `myesspages`.

   If the user has no role, they are redirected to that URL. If the user has a role, they are redirected to `instance.service-now.com/navpage.do`

*Changes to SAML 2.0 and digest token configuration*

Multiple provider single sign-on allows administrators to configure SAML 2.0 Update 1 and digest token as authentication methods.

Multiple provider single sign-on should be activated before you configure your SAML 2.0 Update 1 and digest token properties. After you request and activate multi-provider SSO, you must then set it up. After setting up multi-provider SSO, you can create or update the SAML 2.0 Update 1 and digest token configurations. You can use either or both authentication solutions with multi-provider SSO.

**Note:** The Integration - Multiple Provider Single Sign-On Installer plugin removes the SAML application from the navigator. The necessary SAML settings are migrated to the Multi-Provider SSO application into the SAML2 Migrated table. You can still modify items like the x509 certificate, IdP details, and so on through the Multi-Provider SSO application.
SAML 2.0

The Security Assertion Markup Language (SAML) is an XML-based standard for exchanging authentication and authorization data between security domains.

SAML exchanges security information between an identity provider (a producer of assertions) and a service provider (a consumer of assertions). SAML is a product of the OASIS Security Services Technical Committee. When implemented correctly, SAML is one of the most secure methods of single sign-on available.

The SAML 2.0 integration enables single sign-on by exchanging XML tokens with an external Identity Provider (IdP). The identity provider authenticates the user and passes a NameID token to the system. If the system finds a user with a matching NameID token (for example, the email address), the instance logs in that user.

If you are using the SAML 2.0 plugin for Single Sign-on authentication, then you need to set the glide.ui.rotate_sessions property to false. Otherwise, it interferes with the session information sharing that takes place between the instance and the Identity Provider. Users with the security_admin elevated privilege can access this high security property by selecting System Security > High Security Settings.

**Note:** It is recommended that customers using an existing SAML 2.0 integration upgrade to the latest SAML 2.0 integration update.

SAML concepts

Familiarize yourself with these SAML concepts.

*Typical SAML process flow (diagram)*

A typical SSO logic flow involves looking for an active session, checking user credentials, and creating the necessary token.
Figure 525: SSO Standard

Login (AuthnRequest) process flow

SAML 2.0 specifies a Web Browser SSO Profile that involves exchanging information among an identity provider (IdP), a service provider (SP), and a principal (user) on a web browser.

The identity provider can be any SSO service offering SAML authentication services (for example SSOCircle). The service provider is always an instance. The message flow begins with a request for a secured resource at the service provider.

Request the target resource at the SP

The principal requests a target resource at the service provider:

https://instance.service-now.com/
The instance checks the request to see if the SAMLRequest and RelayState URL parameters are present. If they exist, the user has already validated with the IdP and can skip steps 2–6.

**Issue AuthnRequest to Identity Provider**

The instance constructs an AuthnRequest to be sent to the IdP using the SAMLRequest value. The instance also constructs and sends a RelayState URL parameter value.

The RelayState token is an opaque reference to state information maintained at the service provider. The value of the SAMLRequest parameter is the deflated and base64 encoded value of the `<samlp:AuthnRequest>` element:

```
```

The integration then URL-encodes the `<samlp:AuthnRequest>` element and sends it as the SAMLRequest URL parameter.

The SSO service processes the `<samlp:AuthnRequest>` element by URL-decoding, base64-decoding and inflating the request, in that order. It then performs a security check. If the user does not have a valid security context, the IdP identifies the user by prompting for login credentials. If the user is already logged in, the IdP simply responds with the SAMLResponse and RelayState URL parameters (see step 3).

**Respond with an SAMLResponse and RelayState**

After collecting the required login credentials, the SSO service validates the request and responds with a document containing an XHTML form:

```
<form method="post" action="https://instance.service-now.com/navpage.do" ...><input type="hidden" name="SAMLResponse" value="response ..." /><input type="hidden" name="RelayState" value="token ..." />
...
<input type="submit" value="Submit" /></form>
```

The value of the RelayState parameter comes from this step. The value of the SAMLResponse parameter is the base64 encoding of the following `<samlp:Response>` element:

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 ID="s2cdc74f37f923e26felaeeec42b70a93d24230334f"
 InResponseTo="90AA6073F01567BFB0DF194F596314E2" Version="2.0"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="s23e536bc51b84874d32999dec162d9c2e338823b"
```

© 2017 ServiceNow. All rights reserved. 1906
Validate SAMLResponse

The SAMLResponse value is base64 decoded and inflated to reveal the XML document in step 3. The login script extracts the XML value from the //Subject/NameID element and uses it to look up an existing user in the User table.

The login script also extracts the session ID from the //AuthnStatement/@SessionIndex element and stores it for the LogoutRequest.

Logout (LogoutRequest) process flow
During logout, the instance issues the SAML 2.0 LogoutRequest service call to the IdP.

This service logs the user out and then redirects back to the specified logout URL.
Figure 526: SAML 2 Logout

**User Clicks the Logout Button**

The user clicks the **Logout** button and the instance executes the logout script.
LogoutRequest issued

The logout script constructs a SAML 2.0 LogoutRequest and posts it to the preconfigured SingleLogoutRequest SAML 2.0 service at the IdP. The IdP deflates the request and then base64 encodes it. An example LogoutRequest looks like this:

```xml
```

User Logs Out

The user logs out of the IdP. The IdP redirects back to the instance, which in turns redirects back to the IdP since the user is not logged in.

URL information for an SSO provider

During a login challenge resulting from a URL link into the instance that requires an SSO session, the referring URL might need to be supplied to the SSO provider so that after authentication, the URL can be passed back to the instance and linked to the correct resource.
Figure 527: SSO Target Redirect

Installation exit return values have been enhanced to pass a URL instead of, or in addition to the URL defined by the properties. Usually, you would return a username or a predefined string value to control
authorize or challenge the SSO session. The following examples show the extended behavior of passing a URL.

```
return "failed_missing_requirement:%26amp;TARGET=https://instance.service-now.com/nav_to.do?uri=incident.do?sys_id=12345";
```

The example above passes the URL `https://instance.service-now.com/nav_to.do?uri=incident.do?sys_id=12345` to the SSO provider in the form of a URL parameter named TARGET.

**Note:** It is assumed that the SSO provider will use that information in the TARGET parameter to redirect back to the instance when the user credentials have been collected and authentication passed.

A colon : demarcates the two return values and an encoded & (%26amp;) concatenates the URL defined in the property glide.authenticate.failed_missing_requirement and the TARGET parameter.

SAML 2.0 Single Sign-On - Update 1

The SAML 2.0 Single Sign-On - Update 1: security enhancements plugin improves integration security by requiring additional checks against the SAMLResponse URL parameter.

The integration explicitly checks the SAML response for the proper Identity Provider (IdP) and intended audience URLs.

**Additional SAML response validations**

With Update 1, the integration validates these elements in the SAMLResponse.

- An Issuer element that matches the value listed in the issuer system property
- The SubjectConfirmation and SubjectConfirmationData elements with a Recipient attribute
- The AudienceRestriction and Audience elements that match the value listed in the audience system property

**Support for Signed SingleLogoutRequest**

With Update 1, the SAML 2.0 integration has the option to sign SingleLogoutRequest elements. Some IdPs, such as Microsoft ADFS, require a signed SingleLogoutRequest.

**Support for AuthnContextClass**

With Update 1, the SAML 2.0 integration has the option to specify the method by which the IdP authenticates the user in the AuthnContextClass element. For example, the integration can now specify contexts such as form-based Password Protected Transport or Kerberos. See *(Optional) Enable Providing an Authentication Context Class* for instructions on setting an authentication context class.

**Properties**

The SAML 2.0 Update 1 plugin includes the following system properties.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Identity Provider URL which will issue the SAML2 security token with user info. glide.authenticate.sso.saml2.idp</td>
<td>Enter the value of the Issuer element that the integration uses to validate the IdP URL.</td>
</tr>
<tr>
<td>Sign LogoutRequest. Set this property to true if the Identity Provider's SingleLogoutRequest service requires signed LogoutRequest. glide.authenticate.sso.saml2.require_signed_logoutrequest</td>
<td>Select whether the IdP requires a signed logout request.</td>
</tr>
<tr>
<td>Select whether the IdP requires a signed logout request. glide.authenticate.external.logout_redirect</td>
<td>Enter the URL where the integration redirects users after they log out. Typically, you set this property to a UI page if you are using Kerberos authentication to prevent users from being redirected back to the IdP and logging in again after a logout request.</td>
</tr>
<tr>
<td>The audience uri that accepts SAML2 token. (Normally, it is your instance URI. For example: https://&lt;instance name&gt;.service-now.com.) glide.authenticate.sso.saml2.audience</td>
<td>Enter the value of the Audience element that integration uses to validate the SP URL in the SAMLResponse.</td>
</tr>
<tr>
<td>Create an AuthnContextClass request in the AuthnRequest statement. glide.authenticate.sso.saml2.createrequestedauthncontext</td>
<td>Select whether to create an AuthnContextClass element in the SAMLRequest that specifies the login mechanism the IdP should use to authenticate the user. Not all IdPs support a AuthnContextClass element in the SAMLRequest. If you select Yes, you must specify the URN of the context class with the glide.authenticate.sso.saml2.authncontextclassref property.</td>
</tr>
<tr>
<td>The AuthnContextClassRef method that we will request in our SAML 2.0 AuthnRequest to the Identity Provider glide.authenticate.sso.saml2.authncontextclassref</td>
<td>Enter the URN of the login mechanism you want the IdP to use to authenticate users. For example, by default the system uses the forms-based Password Protected Transport authentication context urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport.</td>
</tr>
<tr>
<td>The alias of key entry stored in SAML 2.0 SP Keystore used to sign SAML 2 requests. glide.authenticate.sso.saml2.signing_key_alias</td>
<td>Enter the alias of the key that signs SAML 2 logout requests. You will have to create a Java Keystore for the alias.</td>
</tr>
<tr>
<td>The password of key entry stored in SAML 2.0 SP Keystore used to sign SAML 2 requests. glide.authenticate.sso.saml2.signing_key_password</td>
<td>Enter the password for the key that signs SAML 2 logout requests.</td>
</tr>
<tr>
<td>The number in seconds before &quot;notBefore&quot; constraint, or after &quot;notOnOrAfter&quot; constraint, to consider still valid. glide.authenticate.sso.saml2.clockskew</td>
<td>Enter the number of seconds between the two attributes that make up the SAMLResponse nonce. A valid SAMLResponse must fall between the notBefore and notOnOrAfter date-time values.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>AuthnRequest URL for eSignature Authentication. com.snc.integration.saml_esig.idp_authnrequest_url</td>
<td>Enter the URL that points to the SAML 2.0 Identity Provider AuthnRequest Consumer for eSignature Authentication. In most cases, this will be the same as the AuthnRequest URL used in general authentication. Leave this setting blank if you intend to use the same AuthnRequest Consumer URL that is used for general SAML 2.0 authentication in your instance.</td>
</tr>
<tr>
<td>The SAML 2.0 Assertion Consumer URL for eSignature authentication. com.snc.integration.saml_esig.approval_consumer_url</td>
<td>In most cases, this URL will be: <a href="https://YOURINSTANCE.service-now.com/consumer.do">https://YOURINSTANCE.service-now.com/consumer.do</a>. However, if you employ a customized method of handling the SAML authentication for eSignature, you can set up your own consumer URL.</td>
</tr>
<tr>
<td>The SAML 2.0 Assertion Consumer Index for eSignature authentication. com.snc.integration.saml_esig.assertion_consumer_service_index</td>
<td>If your Service Provider has more than one URL set for the AssertionConsumerURL, you can set the index to use for eSignature, starting with index 1 or more.</td>
</tr>
<tr>
<td>Authentication Pop-up Dialog Width. com.snc.integration.saml_esig.popup_dlg_width</td>
<td>When a user approves a request using eSignature, a dialog allows the user to enter their credentials. This setting controls the width of that dialog box.</td>
</tr>
<tr>
<td>Authentication Pop-up Dialog Height. com.snc.integration.saml_esig.popup_dlg_height</td>
<td>When a user approves a request using eSignature, a dialog allows the user to enter their credentials. This setting controls the height of that dialog box.</td>
</tr>
</tbody>
</table>

**SAML 2.0 update 1 requirements**

The SAML 2.0 update requires:

- Activating the SAML 2.0 Update 1 plugin
- Additional metadata from the SAML 2.0 Identity Provider (IdP)
  - SAML Request can include an AuthnContextClass element to specify the Service Provider's preferred login mechanism such as form-based authentication or Kerberos. If this element is not specified, the IdP chooses the login method.
  - SAML Response must include an Issuer element that matches the value listed in the issuer system property
  - SAML Response must include SubjectConfirmation and SubjectConfirmationData elements with a Recipient attribute
  - SAML Response must include AudienceRestriction and Audience elements that match the value listed in the audience system property

**SAML 2.0 integration requirements**

The SAML 2.0 integration requires several items, such as an IdP that provides a certificate and a single logout service.
• Activating the latest SAML 2.0 plugin
• Access to a SAML 2.0 Identity Provider (IdP)
  • Must provide an authentication request service
  • Must provide a single logout service
  • Must have a valid certificate
  • Must accept Service Provider (SP) metadata
  • Must use the NameID element in the SAMLResponse

Upgrade from previous versions

The following table lists the actions to take if you are running a previous versions of SAML.

<table>
<thead>
<tr>
<th>Previous SAML version</th>
<th>Action required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML 1.1</td>
<td>Contact Customer Support to migrate your instance to SAML 2.0 Update 1.</td>
</tr>
<tr>
<td>SAML 2.0</td>
<td>Update to SAML 2.0 Update 1.</td>
</tr>
</tbody>
</table>

SAML setup

SAML 2.0 setup involves several steps, including configuring IdP settings and installing the certificate.  
*Activate SAML 2*

SAML 2.0 is activated as part of the Integration - Multiple Provider Single Sign-On Installer plugin, which can be activated by an administrator.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

*Identity Provider (IdP) system properties*

An IdP generally offers an XML document containing their authentication and logout metadata.

For example, **SSOCircle** publishes their metadata online.

Browse the IdP metadata to find these entries:

• The **SingleSignOnService** element with a **Binding** attribute that contains a value of HTTP-Redirect. The **Location** attribute lists the URL the integration requires for the AuthnRequest service.
• The **SingleLogoutService** element with a **Binding** attribute that contains a value of HTTP-Redirect. The **Location** attribute lists the URL the integration requires for the SingleLogoutRequest service.
Note: The SAML 2.0 integration only supports binding to IdP services by HTTP-Redirect.

For example:

```
```

```
```

Figure 528: IdP properties

Set the IdP issuer URL

Provide the URL to the IdP's who will issue the security token.

Role required: admin

The integration verifies that each SAML response contains the same URL listed in this system property as the URL listed in the Issuer element. For example:

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    Destination="https://demoi2.service-now.com/navpage.do"
    ID="s28da6774c88ae1eab292bf25fe625db81919d8e1e"/>
```
1. Navigate to SAML 2 Single Sign-on > Properties.

2. In the property The Identity Provider URL which will issue the SAML2 security token with user info., enter the URL to your IdP. By default, the integration contains the URL to SSOCircle http://idp.ssocircle.com.

Set the AuthnRequest service URL
Using the IdP's metadata, set the request service URLs for the integration's IdP.

Role required: admin

1. In the property The base URL to the Identity Provider's AuthnRequest service. The AuthnRequest will be posted to this URL as the SAMLRequest parameter, enter the URL to the HTTP-Redirect binding obtained from the SingleSignOnService element.

2. Select the check box next to Sign AuthnRequest to enable the Identity Provider's single-sign on service to receive a signed AuthnRequest.

3. In the property When SAML 2.0 single sign-on fails because the session is not authenticated, or this is the first login, redirect to this URL. This is the base URL where the initial SAML 2.0 AuthnRequest is sent using the SAMLRequest parameter, enter the URL to the HTTP-Redirect binding obtained from the SingleSignOnService element.

By default, the integration contains the URL to the SSOCircle service.

Set the SingleLogoutRequest service URL
Using the IdP's metadata, set the request service URLs for the integration's IdP.

Role required: admin

1. In the following property, enter the URL obtained from the SingleLogoutService element: The base URL to the Identity Provider's SingleLogoutRequest service. The LogoutRequest will be posted to this URL as the SAMLRequest parameter. The LogoutRequest is posted to this URL as the SAMLRequest parameter.

By default, the integration contains the URL to the SSOCircle service.

2. In the property URL to redirect users after logout, typically back to the portal that enabled the SSO (e.g. http://portal.companya.com/logout), enter the URL where you want to redirect users after they successfully logout. If your IdP uses form-based authentication, enter the URL to your IdP’s login form. If your IdP uses a non-form-based authentication method such as Kerberos, you should set the URL to a static logout page. This way, users who log out do not get immediately get redirected to the IdP and login again.

By default, the integration contains the URL to the static UI page external_logout_complete.do.

(Optional) Enable signed logout requests
Some IdPs require the Service Provider to sign logout requests with a certificate.

Role required: admin
If your IdP requires signed logout requests, use the IdP's metadata to set the following system properties.

1. From the property Sign LogoutRequest. Set this property to true if the Identity Provider’s SingleLogoutRequest service requires signed LogoutRequest, select Yes to specify that your IdP requires a signed logout request, or select No to use unsigned logout requests.

2. If you selected Yes to Sign LogoutRequest, then in The protocol binding for the Identity Provider's SingleLogoutRequest service. (Value can be either "urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" or "urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST").) property, enter the one of the supported values listed in Binding attribute from the SingleLogoutService element.

   By default, the integration uses an HTTP-Redirect binding.

3. Click Update.

4. Install a Service Provider (SP) key store.

Examples of third-party SAML identity provider configurations

These documents provide examples of possible IdP setup configurations.

While the instance does not typically provide instructions for configuring third-party SAML IdP products, customers occasionally provide examples of how they have implemented their SAML IdP with the instance.

**Note:** The instance does not provide support for these example configurations.

<table>
<thead>
<tr>
<th>Identity Provider</th>
<th>Example Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Active Directory Federation Services 2.0</td>
<td>Configuring ADFS 2.0 to Communicate with SAML 2.0</td>
</tr>
<tr>
<td>SSOCircle</td>
<td>SSOCircle (Video Tutorial)</td>
</tr>
</tbody>
</table>

_Service Provider (SP) system properties_

These system properties define how the instance interacts with the IdP as a Service Provider.
**Service Provider (Service-Now) properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The URL to the Service-now instance homepage.</td>
<td><a href="https://yourinstance.service-now.com/navpage.do">https://yourinstance.service-now.com/navpage.do</a></td>
</tr>
<tr>
<td>The entity identification, or the issuer</td>
<td><a href="https://yourinstance.service-now.com">https://yourinstance.service-now.com</a></td>
</tr>
<tr>
<td>The audience uri that accepts SAML2 token. (Normally, it is your instance URL. For example: <a href="https://demo.service-now.com">https://demo.service-now.com</a>.)</td>
<td><a href="https://yourinstance.service-now.com">https://yourinstance.service-now.com</a></td>
</tr>
<tr>
<td>The User table field to match with the Subject's NameID element in the SAMLResponse</td>
<td>email</td>
</tr>
<tr>
<td>The NameID policy to use for returning the Subject's NameID in the SAMLResponse. Your SAML identity provider will have to support this by declaring the policy in its metadata. The NameID value is used to match with the specified field in the User table to lookup the user.</td>
<td>um:oasis.names.tc:SAML:1.1:nameid-format:emailAddress</td>
</tr>
</tbody>
</table>

Create an AuthnContextClass request in the AuthnRequest statement. This tells the IdP that ServiceNow requires that they present a specific login mechanism such as a form, Kerberos, etc. If the AuthnRequest doesn't specify an AuthnContextClass, the IdP will choose the most appropriate method.

- Yes | No

The AuthnContextClassRef method that we will be included in our SAML 2.0 AuthnRequest to the Identity Provider

- um:oasis.names.tc:SAML:2.0:ac.classes:PasswordProtectedTransport

The alias of key entry stored in SAML 2.0 SP Keystore used to sign SAML 2 requests.

- saml2sp

The password of key entry stored in SAML 2.0 SP Keystore used to sign SAML 2 requests.

- ********

---

**Figure 529: Service Provider properties**

Set the instance URL for SAML.
Set the instance-specific URLs in SAML system properties.
Role required: admin

1. In the property The URL to the Service-now instance (usually this instance), enter the URL (including login page) of the instance for which the IdP authenticates. For example: https://yourinstance.service-now.com/navpage.do

2. In the property The entity identification, or the issuer, enter the base URL (excluding login page) of the instance for which the IdP authenticates. For example: https://yourinstance.service-now.com/

Set the audience URL for SAML
The Audience system property allows your instance to verify that it is the intended recipient of a SAML response.

Role required: admin

The integration verifies that each SAML response contains the same URL listed in this system property as the URL listed in the Audience element. For example:

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
ID="s2cdc74f37f923e26fe1aeec42b70a93d24230334f"
InResponseTo="90AA6073F01567BF0DF194F596314E2" Version="2.0"
...
<saml:Conditions NotBefore="2012-01-30T19:57:10Z"
...
</samlp:Response>
```

1. Navigate to SAML 2 Single Sign-on > Properties.
2. In the property The audience uri that accepts SAML2 token. (Normally, it is your instance URI. For example: https://<instance name>.service-now.com.), enter the URL of your instance. For example, https://demoi2.service-now.com. This URL must match the value of the Audience element in the SAML Response.
3. Click Update.

Set up a NameID policy for SAML
SAML 2.0 requires the IdP to exchange a NameID token with the service provider.

Role required: admin

For the SAML 2.0 integration the NameID token must map to a particular field in the User table. The integration uses the NameID token's value to determine what user the IdP authenticates.

1. Browse the IdP metadata to find the NameIDFormat element that contains a value of emailAddress. The value of this element is the default format that the integration uses.
2. Review other NameIDFormat elements to determine if there are formats that match other fields in the User table.

Determine what User table field matches the NameID token
Identity providers specify what format the NameID token has.

Role required: admin

Setting up SAML 2.0 requires selecting a field from the User table that matches the format of the NameID token. Typically, IdPs offer the option to use an email address as the NameID token. Since the User table contains an email field, this field is a logical choice for use as a NameID token. To use another field from
the User table as the NameID token, first verify that the IdP offers a NameID format that matches the value of a User table field. This may require adding the field to the User table.

1. Compare the available formats in the IdP’s NameIDFormat element to fields in the User table.
2. Select a NameID format where there is a matching value in the User table.
3. In the The User table field to match with the Subject's NameID element in the SAMLResponse field, enter the name of the User table field to search for matching values in the NameID token.

By default, the integration uses the email field.

Set the IdP NameID policy
Specify what format the IdP uses for the NameID token.
Role required: admin
This format is listed as part of the IdP’s metadata.

1. In the property The NameID policy to use for returning the Subject’s NameID in the SAMLResponse. Your SAML identity provider will have to support this by declaring the policy in its metadata. The NameID value is used to match with the specified field in the User table to lookup the user., enter the value of the NameIDFormat element the integration uses.

By default, the integration uses the SSOCircle NameIDFormat for email addresses.

2. Click Save.

Values in the User table field for SAML
Ensure that the integration’s User table field contains appropriate matching values.

For example, if the integration uses the email field as the NameID token, ensure that the instance lists the same email address as the IdP. The integration fails to authenticate any user who does not have a matching value for the NameID token.

(Optional) Enable providing an authentication context class for SAML
You can enable the instance to send an authentication context class request to the IdP containing your instance’s preferred authentication request format.
Role required: admin
If you enable creating an AuthContextClass message, then you must also specify an authentication context class reference format.

**Note:** Some IdP’s do not allow the Service Provider to set the authentication context class. Disabling this setting allows the IdP to choose the authentication context class.

1. From the property Create an AuthnContextClass request in the AuthnRequest statement, select Yes to specify a particular context class such as Password Protected Transport, or select No to have the IdP select the most appropriate context class.
2. If you selected Yes to Create an AuthnContextClass request in the AuthnRequest statement, then in The AuthnContextClassRef method that we will request in our SAML 2.0 AuthnRequest to the Identity Provider property, enter the URN of the context class you want to use for authentication (see table).

<table>
<thead>
<tr>
<th>Authentication type</th>
<th>Authentication context class URN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forms-based authentication</td>
<td>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</td>
</tr>
<tr>
<td>Kerberos-based authentication</td>
<td>urn:federation:authentication:windows</td>
</tr>
</tbody>
</table>
By default, the integration uses a Password Protected Transport authentication method.

3. Click **Update**.

(Optional) Set keystore properties for signing logout requests for SAML

The Keystore properties allow the integration to sign logout requests using your signed server and signed CA certificates.

Role required: admin

1. In the property The alias of key entry stored in SAML 2.0 SP Keystore used to sign SAML 2 requests, enter the alias name you created for the SAML 2.0 Keystore. By default, the integration looks for the alias saml2sp.
2. In the property The password of key entry stored in SAML 2.0 SP Keystore used to sign SAML 2 requests, enter the password to your SAML 2.0 Keystore. By default, the password is the same as the default alias name.
3. Click **Update**.
4. Regenerate your **SP metadata**.

Install a service provider keystore for signing SAML requests

Use the following steps to remove the existing example key store and install your own Service Provider key store containing your public and private key pair.

Role required: admin

1. Create a Service Provider key store.
2. Navigate to **SAML 2 Single Sign-on > Certificate**.
3. Click **SAML 2.0 SP Keystore**.
4. Click the **Manage Attachments** link.
5. Select the Delete checkbox next to saml2sp_keystore.
6. Click **Remove**.
7. Click **Choose Files** and select the Keystore containing your signed certificates.
8. Click **Attach**.
9. Close the Attachments popup.
10. In Key store password, enter the password to access the SAML 2 alias.
11. Click **Update**.

Create a service provider keystore

In order for your instance to sign logout requests, you must create a Java Key store containing the following items.

- Signed server certificate for the instance
- Signed CA certificate
- Public and private key pair

You may create your own signed certificate with a private certificate authority or purchase one from a public certificate authority.

The following steps illustrate how to generate a new Java Keytool keystore file, create a certificate signing request (CSR), and import certificates. Any root or intermediate certificates need to be imported before importing the primary certificate for your domain. Type these commands in a command line interface.
**Note:** These instructions are not specific to the platform and require technical knowledge of security certificates to complete. Technical Support cannot assist in creating the certificates.

1. Generate a Java keystore and key pair.

   ```
   keytool -genkey -alias mydomain -keyalg RSA -keystore my.keystore
   ```

2. Generate a CSR for an existing Java keystore.

   ```
   keytool -certreq -alias mydomain -keystore my.keystore -file mydomain.csr
   ```

3. Import a root or intermediate certificate authority CA certificate to an existing Java keystore.

   ```
   keytool -import -trustcacerts -alias root -file Thawte.crt -keystore my.keystore
   ```

4. Import a signed primary certificate to an existing Java keystore.

   ```
   keytool -import -trustcacerts -alias mydomain -file mydomain.crt -keystore my.keystore
   ```

*(Optional) Advanced SAML properties*

The following advanced settings allow you to further increase security and debug the integration.

### Advanced settings

- **The number in seconds before "notBefore" constraint, or after "notOnOrAfter" constraint, to consider still valid**
  - 60

- **Turn on debug logging for SAML 2.0 Authentication**
  - Yes | No

Figure 530: Advanced settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of seconds &quot;notBefore&quot; constraint, or after &quot;notOnOrAfter&quot; constraint, to consider still valid</td>
<td>Enter the number of seconds to add to the NotBefore and NotOnOrAfter constraints to account for time differences between the IdP clock and SP clock. These constraints prevent against replay attacks by denying requests that are not made within the specified time frame. If the IdP clock and SP clock are significantly different, network latency may result in the SAML request being unauthorized. This property adds a grace period during which SAML requests and responses are still considered valid.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Turn on debug logging for SAML 2.0 Authentication</td>
<td>Select <strong>Yes</strong> to enable additional logging information for SAML 2.0 events.</td>
</tr>
</tbody>
</table>

**Install the identity provider certificate**

You can paste a PEM certificate into a X.509 Certificate form so the identity provider can verify communications with the service provider.

Role required: admin

The IdP's certificate is located within the IdP's metadata. The IdP developer determines where the certificate metadata resides when creating the local IdP.

**Tip:** Make sure that the **Certificate Expiring** and **Certificate Expired** notifications are enabled so you remember to update the certificate.

**Note:** Certificates for single-sign on should always be in PEM format to work with SAML certificates.

1. Navigate to **SAML Single Sign-on > Certificate**.
2. Fill in the form fields (see table).
3. Click **Save**.
**Note:** The integration does not currently sign the certificate in communications between the instance and the IdP.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The certificate name. Do not change the Name entry. The name of the X.509 certificate must be SAML 2.0 in order for the integration to use it. This requirement is only true if you are not using Multiple provider single sign-on on page 1891.</td>
</tr>
<tr>
<td>Expiration notification</td>
<td>Select this option to send a notification to the users selected in the <strong>Notify on expiration</strong> field. By default, this is enabled.</td>
</tr>
<tr>
<td>Notify on expiration</td>
<td>Select the users to revive the notification regarding certificate expiration. If no users are selected, the logged in user is added by default, along with the last two logged in users with the administrator role.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Warn in days to expire</td>
<td>The number of days before expiration that the instance send the notification. Enter a value of at least 20.</td>
</tr>
<tr>
<td>Active</td>
<td>A check box to indicate that this certificate is active.</td>
</tr>
<tr>
<td>Format</td>
<td>A PEM or DER certificate. SAML uses PEM format.</td>
</tr>
<tr>
<td>Type</td>
<td>The certificate container. The instance recognizes certificates from trust stores, Java keystore, and PKCS#12 keystores.</td>
</tr>
<tr>
<td>Valid from</td>
<td>The instance automatically adds the certificate valid from date to this field. Attach the certificate to the X.509 certificate record to populate this field.</td>
</tr>
<tr>
<td>Expires</td>
<td>The instance automatically adds the certificate expiration date to this field. Attach the certificate to the X.509 certificate record to populate this field.</td>
</tr>
<tr>
<td>Expires in days</td>
<td>The calculated number of days to expiration.</td>
</tr>
<tr>
<td>Short description</td>
<td>A description for the certificate.</td>
</tr>
<tr>
<td>Issue</td>
<td>The instance automatically adds the certificate issuer to this field. Attach the certificate to the X.509 certificate record to populate this field.</td>
</tr>
<tr>
<td>Subject</td>
<td>The instance automatically adds the certificate subject to this field. Attach the certificate to the X.509 certificate record to populate this field.</td>
</tr>
<tr>
<td>PEM Certificate</td>
<td>Enter the value of the X509 certificate.</td>
</tr>
</tbody>
</table>

Replacing a missing SAML certificate
If the Certificate module displays a blank page, the SAML 2.0 certificate record has been deleted. You can replace the missing certificate by manually creating a certificate record.

Role required: admin

1. Navigate to System Definition > Certificates.
2. Create a new record called SAML 2.0.

**Important:** You MUST use this name.

3. Click SAML 2 Single Sign-on > Certificate.
4. In the PEM Certificate field, enter the value of the ds:X509Certificate element from your IdP’s metadata.
5. Click Save.

Add a Java keystore for SAML
You can add Java keystores to the SAML application if you want another repository for your SAML security certificates.

Role required: admin

By default, SAML 2 Single Sign-on provides a default keystore named **SAML 2.0 SP Keystore**. This keystore is active by default. You can add keystores as needed and specify which one to use by default with a property.

1. Navigate to **System Definition > Certificates**.
2. Fill in the fields on the form (see table).
3. Click **Submit**.

### Table 457: X.509 Certificate form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>Active</td>
<td>Select <strong>Active</strong> to keep the key store available.</td>
</tr>
<tr>
<td>Short description</td>
<td>Enter a description for the key store.</td>
</tr>
<tr>
<td>Type</td>
<td>Select <strong>Java Key Store</strong>.</td>
</tr>
<tr>
<td>Key store password</td>
<td>Enter the password for the key store.</td>
</tr>
</tbody>
</table>

4. To set the default key store, enter `sys_properties.list` in the application filter.
   The list of system properties opens.
5. Find and open the following property: `glide.authenticate.sso.saml2.keystore`.
Geneva ServiceNow ServiceNow Platform

The Value field shows the Sys ID of the default key store: **SAML 2.0 SP Keystore**. Use this property when the system has multiple key stores for signed authentication or signed logout. Starting with Geneva Patch 7, the glide.authenticate.sso.saml2.keystore property uses your configured key store if you upgrade from a version prior to the Geneva release. Prior to the Geneva Patch 7, this property kept the default value upon upgrade, regardless of any changes you made.

6. Paste the Sys ID of the key store record that you created into the **Value** field.
7. Click **Update**.

*Generate instance service provider (SP) metadata*

After setting all the integration properties, generate the instance SP metadata.

Role required: admin

The IdP needs the instance SP metadata to authenticate and forward requests.

1. Navigate to **SAML 2 Single Sign-on > Metadata**. The integration automatically generates the instance's SP metadata from the system property settings.
2. Copy the SP metadata in the text box. For example:

   ```xml
   <EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
     entityID="https://demoi2.service-now.com"/>
     <SPSSODescriptor
     AuthnRequestsSigned="false" WantAssertionsSigned="false"
     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
     <SingleLogoutService
     Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
     Location="https://demoi2.service-now.com/navpage.do" />
     <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
     NameIDFormat><AssertionConsumerService isDefault="true" index="0"
     Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
     Location="https://demoi2.service-now.com/navpage.do"/>
     SPSSODescriptor</EntityDescriptor>
   
   3. Provide the instance SP metadata to the IdP. For example, SSOCircle allows a user to provide the SP metadata online.

*Test the SAML integration*

After completing all other setup tasks, test the integration.

Role required: admin

1. Log in to the instance as a user with the admin role.
2. Navigate to **SAML 2 Single Sign-on > Properties**.
3. In the property **Enable external authentication**, select **Yes**.

**Note:**

Enabling external authentication requires all users to use SAML 2.0 single sign-on. Thus, if anyone tries to access the application in an unauthenticated state, the instance automatically sends an authentication request to the (IdP) and redirects the user to the SAML IdP Authentication page.

4. Click **Save**.
5. Log out of the instance.
6. Browse to the instance URL. If the integration is functioning properly, the IdP should ask for the users credentials.

List of common SAML login errors

Use this list of common login errors and their solutions to troubleshoot your SAML integration.
### Table 458: Troubleshooting login errors table

<table>
<thead>
<tr>
<th>Error or symptom</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication fails and the login request generates an infinite loop between the system and the IdP.</td>
<td>Set (or create) the system property <code>glide.authenticate.failed_redirect</code> to redirect failed authentication requests to this URL. Typically the URL endpoint is an error page or logout page.</td>
</tr>
<tr>
<td>Login requests generate an infinite loop between the system and the IdP when High Security is active.</td>
<td>The High Security plugin's rotating session feature can cause problems with the SAML 2.0 authentication process. Disable the rotating session feature when using SAML 2.0 for authentication. SAML 2.0 needs to redirect URLs to an IdP. When rotating sessions are enabled, any redirection causes the session to rotate and forces the instance to query the IdP again. This causes an endless loop of new sessions between the instance and the IdP.</td>
</tr>
</tbody>
</table>

**Force single sign-on login (SSO) only**

When single sign-on is enabled, you might not want users to see the login page or allow user to login locally.

In other words, if a user attempts to go to https://customerX.service-now.com, the internal company portal should be displayed instead of the default login page. Likewise, when a user logs out of the application, the browser should redirect them to a specific internal page. You can set redirection properties within the instance to ensure that users see the single sign-on page rather than the default login page.

**Redirection properties**

When a user logs out, or if there is a failed attempt to sign on using SSO, you can define where the user is taken next, such as a main portal page or a knowledge base article with SSO login information. Use the following properties to specify the URLs. If one of these properties does not exist in your instance, you can create the property.

- **glide.authenticate.failed_requirement_redirect** URL to redirect users when they attempt to access a page that is private (for example, to view an incident) and do not provide SSO credentials. The property is typically set to a customer's login portal (for example, http://portal.companya.com/).

- **glide.authenticate.failed_redirect** URL to redirect users after a failed SSO attempt. You can redirect to a public knowledge article that describes the error and has helpful links (for example, http://portal.companya.com/error).

- **glide.authenticate.external.logout_redirect** URL to redirect users after logging out, typically back to the portal that enabled the single sign-on log in (for example, http://portal.companya.com/logout).

- **glide.authentication.external.disable_local_login** When set to true, requires SSO credentials for the main login page. Defaults to false. This property needs to be used in conjunction with the
The following table shows the relationship between the Installation Exit return values, the properties, and the expected behavior.

### Table 459: Forcing login using SSO only

<table>
<thead>
<tr>
<th>Return value</th>
<th>Property</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>failed_missing_requirement</td>
<td>glide.authenticate.failed_requirement_redirect</td>
<td>When this value is returned, it indicates that the required SSO credentials are not present in the session. Login fails and the session is redirected to the URL specified by the property. This is usually the URL for the SSO provider where login is challenged and credentials are collected.</td>
</tr>
<tr>
<td>failed_authentication</td>
<td>glide.authenticate.failed_redirect</td>
<td>When this value is returned, it indicates that the supplied SSO credentials failed authentication, the user does not exist, or the user is locked out. Login fails and the session is redirected to the URL specified by the property. This is usually the URL for the SSO provider where login is challenged and credentials are collected.</td>
</tr>
<tr>
<td>&lt;user_id&gt;</td>
<td>N/A</td>
<td>Login authorized for the user specified by <code>&lt;user_id&gt;</code>. This value matches with the field name defined in the SSO property glide.authenticate.header.value (&quot;the instance's field name to match against the incoming header&quot;)</td>
</tr>
</tbody>
</table>

### Restricting local login

As a security precaution, you should do more than rely on redirection properties to prohibit logging in locally. If a user should never log in locally and will always be authenticated by your internal single sign-on system, then a random password should be assigned to each user that is imported into the instance. The random password is most easily set at the time of the user import. If the user data is imported into your system through an import set, you can create an onBefore transform script using the following code.

```javascript
var r = new Packages.java.util.Random();
var str1 = Packages.java.lang.Long.toString(Packages.java.lang.Math.abs(r.nextLong()), 36);
var str2 = Packages.java.lang.Math.abs(r.nextDouble());
```

© 2017 ServiceNow. All rights reserved.
Integrating SAML 2.0 with other features

You can integrate your SAML 2.0 solution with other features like E-Signature, deep linking, and ADFS.

**Add E-Signature support for SAML**

Configure the following properties for E-Signature with SAML 2.0 update 1.

When E-signature is active with Multi-SSO, SAML properties are not used. The system adds E-signature properties to the SAML2 Update1 Properties [saml2_update1_properties] table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assertion Consumer Index for eSignature authentication</td>
<td>An index number that identifies the endpoint.</td>
<td>1</td>
</tr>
<tr>
<td>Assertion Consumer URL for eSignature authentication</td>
<td>The URL that identifies the consumer.</td>
<td><a href="https://yourinstance.service-now.com/consumer.do">https://yourinstance.service-now.com/consumer.do</a></td>
</tr>
<tr>
<td>AuthnRequest URL for eSignature Authentication</td>
<td>The URL for authentication</td>
<td>none</td>
</tr>
</tbody>
</table>

If you are using E-Signature with SAML 1.0 or SAML 2.0 (not including update 1), see the special configuration instructions: *Using E-Signature with Single Sign-On (SSO).*

**Note:** If you are a Life Science Customer using E-Signature, you must deactivate the User self-lockout prevention business rule. See *KB0547061* for more information.

**Add deep linking support for SAML**

Deep linking allows instances to support direct email links to a particular record in the system.

With the SAML 2.0 integration enabled, deep-linking URLs must pass an authentication check before the IdP redirects the user to the originally requested URL. For example, consider an email that contains this URL: https://<instance name>.service-now.com/nav_to.do?url=incident.do?sys_id=46c88ac1a9fe1981014de1c831fbcf6d

The instance sends an authentication request to the IdP and uses the RelayState URL parameter to preserve the originally requested resource (in this case, url=incident.do?sys_id=46c88ac1a9fe1981014de1c831fbcf6d). After the IdP authenticates the user, the instance reads the value of the RelayState URL parameter and redirects the user to the requested resource (if it exists in the instance).

To add support for deep linking verify that the identity provider supports the RelayState URL parameter.

**ADFS integration with SAML 2.0**

SAML 2.0 single sign-on (SSO) supports integration with Microsoft Active Directory Federation Services (ADFS) 3.0.

For information about installing and configuring ADFS, see *Active Directory Federation Services Overview*.

Set up ADFS for SAML
This procedure uses ADFS 2.0 and shows samportal.example.com as the ADFS website. Replace this with your ADFS website address.

Role required: admin

1. Log into the ADFS 3.0 server and open the management console.
2. Right-click Service and choose Edit Federation Service Properties.

3. Confirm that the General settings match your DNS entries and certificate names.
4. Browse to the certificates and export the Token-Signing certificate.
   a) Right-click the certificate and select **View Certificate**.
   b) Select the **Details** tab.
   c) Click **Copy to File**. The Certificate Export Wizard opens.
   d) Select **Next**.
   e) Ensure the **No, do not export the private key** option is selected, and then click **Next**.
   f) Select **DER encoded binary X.509 (.cer)**, and then click **Next**.
   g) Select where you want to save the file and give it a name. Click **Next**.
   h) Select **Finish**. The instance requires that this certificate be in PEM format. You can convert this certificate using client tools or even online tools such as: SSL Shopper.

5. Use the DER/Binary certificate that you just created, and export it in Standard PEM format.
Set up the instance for ADFS

After you set up ADFS 2.0 or 3.0, set up the instance and SAML 2.0 settings to work with ADFS.

Role required: admin

1. If not already active, contact Technical Support to activate the SAML 2.0 Single Sign-On plugin.
2. Configure SAML 2.0 on page 1904, but when you install the IdP certificate, attach the PEM certificate you created when you Set up ADFS for SAML on page 1930.
3. Click Save.
4. Verify that the Issue and Subject fields have values and that there are no errors. If an error occurs, open the saved PEM formatted certificate in Notepad and copy and paste the certificate into the PEM Certificate field.
5. Verify that the SAML2SingleSignon_update1 installation exit is active.
6. Continue the SAML 2.0 configuration.

Note: When a certificate is updated on the ADFS server, you also need to upload an updated certificate to the instance.

Configure an ADFS relying party

At this point you can take the instance metadata and import it into your ADFS server. However, manual configuration of the relying party appears to be easier to implement.

Role required: admin

1. Navigate to SAML 2 Single Sign-on > Properties and verify that the SAML property Sign AuthnRequest (glide.authenticate.sso.saml2.require_signed_authnrequest) is not active. Only keep this property active if your ADFS administrator can verify that you require signed requests.
2. Copy the metadata that you generated through the SAML 2 metadata link and save it to a file.
3. Log into the ADFS server and open the management console.
5. Select Add Relying Party Trust from the top right corner of the window.
6. The add wizard appears.
7. Click Start to begin.
8. Use the Import File option to import the metadata file.
9. Give it a display name such as ServiceNow and enter any notes you want.
10. Select ADFS 3.0 Profile.
11. Do not select a token encryption certificate. It will use the certificate that is defined on the service that has already been exported. Defining a certificate here will prevent proper communication with the instance.
12. Do not enable any settings on the Configure URL.
13. Enter the instance site to which you connected as the Relying Party trust identifier. In this case use https://company.service-now.com and click Add.
14. Permit all users to access this relying party.
15. Click Next and clear the Open the Claims when this finishes check box.
16. Right-click on the relying party trust and select Properties.
17. Browse to the Advanced tab and set the Secure hash algorithm to SHA-1.

Configure ADFS relying party claim rules
Edit the Claim rules to enable proper communication with the instance.

Role required: admin

1. Log into the ADFS server and open the management console.
2. Right-click the relying party trust and select **Edit Claim Rules**.
3. Click the **Issuance Transform Rules** tab.
4. Select **Add Rules**.
5. Select **Send LDAP Attribute as Claims** as the claim rule template to use.
6. Give the claim a name such as **Get LDAP Attributes**.
7. Set the **Attribute store** to **Active Directory**, the **LDAP Attribute** to **E-Mail-Addresses**, and the **Outgoing Claim Type** to **E-mail Address**.
Edit Rule - Get Attribute

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Get Attribute

Rule template: Send LDAP Attributes as Claims

Attribute store:

Active Directory

Mapping of LDAP attributes to outgoing claim types:

<table>
<thead>
<tr>
<th>LDAP Attribute (Select or type to add more)</th>
<th>Outgoing Claim Type (Select or type to add more)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Mail-Addresses</td>
<td>E-Mail Address</td>
</tr>
</tbody>
</table>

---

c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]

=> issue(store = "Active Directory",

© 2017 ServiceNow. All rights reserved.
8. Select Finish.
9. Select Add Rules.
10. Select Transform an Incoming Claim as the claim rule template to use.
11. Give the Claim a name such as Email to Name ID.
12. Set the Incoming claim type to the Outgoing Claim Type in the previous rule. For example, E-Mail Address.
13. Set the Outgoing claim type to Name ID and the Outgoing name ID format to Email.

   Note: These values must match the Name ID policy you define during SAML 2.0 configuration.
14. Select Pass through all claim values.
This claim rule should look similar to the following rule language.

c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"]
15. Click Finish.

Create a SAML logout endpoint

Create a SAML logout endpoint to allow single logout.

Role required: admin

See this article on ADFS signout for more information.

1. Go to ADFS manager > Trust Relationships > Relying Party Trusts > properties.
2. Under the Endpoints tab, click Add.
3. Configure the settings:
   • Endpoint Type: SAML Logout
   • Binding: POST
   • URL: https://myadfsserver.domain.net/adfs/ls/?wa=wsignout1.0

Test the ADFS configuration

Test your ADFS configuration to verify that it is properly functioning as an identity provider.

Role required: admin

1. Open an Internet Explorer browser.
   This page contains a drop down list of all configured Relaying Party Trusts.
3. Select the relaying party associated with your instance.
4. Click Continue to Sign In.
   If you have configured the SAML 2.0 external authentication properly, you should be automatically logged into the instance.

(Workaround) Enable service provider-initiated authentication

A workaround is available if authentication fails because you do not have SAML 2.0 Update 1. This can happen if users attempt to skip IdP authentication and navigate directly to the instance.

Role required: admin

This is an error with the instance not providing ADFS with the needed definition and semantics for the SPNameQualifier attribute in the SAMLResponse.

To enable service provider-initiated authentication, do one of the following

• Upgrade to SAML 2.0 Update 1 and clear the option to create an AuthnContextClass request. See SAML setup on page 1914 for more information.
• Modify the SAML2 script include to comment out the definitions of the SPNameQualifier attribute when you have SAML 2.0 active (not SAML 2.0 Update 1).
Comment out these lines in the createNameID and createNameIDPolicy functions:

```java
//nid.setSPNameQualifier (serviceURL ) ;
//nameIdPolicy. setSPNameQualifier (serviceURLStr ) ;
```

If you do not want the login prompt from your ADFS server to appear when you access the instance, set the following SAML 2.0 Update 1 property to false: **Create an AuthnContextClass request in the AuthnRequest statement**(glide.authenticate.sso.saml2.createrequestedauthncontext).

(Workaround) Support Kerberos authentication
A workaround is available for the SAML 2.0 integration that changes the authentication context from forms-based authentication to Windows-based authentication.

Role required: admin
Currently, the SAML 2 integration uses a PasswordProtectedTransport or "forms-based authentication" authentication context. This authentication context requires the IdP to present users with a form for authentication credentials. With Kerberos, a SAML session is already active through an established Windows login, so the user does not need to authenticate with the IdP.

1. Navigate to **Multi-Provider SSO > Identity Providers**.
2. Open the **SAML2 Update1** IdP record.
3. Select the **Create an AuthnContextClass request in the AuthnRequest statement** checkbox.
4. Set the **The AuthnContextClassRef method that we will be included in our SAML 2.0 AuthnRequest to the Identity Provider** to one of the following:

   **Table 460: AuthnContextClassRef method values**

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport (Default)</td>
</tr>
<tr>
<td>urn:federation:authentication:windows</td>
</tr>
</tbody>
</table>

5. Click **Update**.

**Email links with SSO**
When using External Authentication (or SSO) that requires URL parameter additions, you must establish how to handle links in email notifications.

The default links contain a URL that directs you to a specific location in the instance, like an Incident or Change Request, without incorporating SSO credentials. Below are examples for directing the user to the location in the instance without logging in on the instance login page.

- **Unencrypted HTTP technique to connect to the /demo instance (it does not navigate to specific record):**
  ```
  https://<instance name>.service-now.com/?
  SM_USER=user_name&DE_USER=lQjIVp7aRJtyPx5+2O/vgU24tbE=
  ```

- **Link (in an email notification) to a specific record, so that the user first goes to the company's own login portal:**
  ```
  https://login.company_portal_page.com/nav_to.do?uri=incident.do?
  sys_id=009f8eda0a0a0b2b01ab4eb094223466%26sysparm_stack=incident_list.do
  %3Fsysparm_query=active=true
  ```

You must set the glide.email.override.url property in your instance to contain the URL of the company portal page. If this property does not exist, you can create it.
• The company portal must then take that URL and construct the redirect URL to the instance preserving the segment necessary to access the specific record, and adding the SSO credentials to the end of the URL:

```
https://<instance name>.service-now.com/nav_to.do?uri=incident.do?
sys_id=009f8eda0a0a0b2b0ab4eb094223466%26sysparm_stack=incident_list.do
%3Fsparm_query=active=true&SM_USER=user_name&DE_USER=lQjIVp7aRJtyPx5+2O/vgU24tbE=
```

Migrating an existing SAML 1.1 integration to SAML 2.0

To migrate from a SAML 1.1 integration to a SAML 2.0 integration, contact customer support.

Update existing SAML 2.0 integration

Perform these steps to update your existing SAML 2.0 integration.

Role required: admin

Request the SAML 2.0 Update 1 Plugin

Contact customer support to request the SAML 2.0 Single Sign-On - Update 1: security enhancements plugin. The plugin applies updated versions of the SAML2SingleSignon installation exit (login script), SAML2Logout installation exit (logout script), and SAML2 script include (script object). See Activate a plugin on page 1233.

Merge Customizations from Existing Installation Exit Scripts into New Scripts

The update saves an inactive copy of the integration’s original installation exit scripts. You can use these copies to merge any customizations you made to the login and logout scripts to the new versions of these installation exits.

### Table 461: Merge Customizations from Existing Installation Exit Scripts into New Scripts

<table>
<thead>
<tr>
<th>Original Installation Exit Script Name</th>
<th>Original Script Status</th>
<th>New Installation Exit Script Name</th>
<th>New Script Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML2SingleSignon</td>
<td>Inactive</td>
<td>SAML2SingleSignon_update1</td>
<td>Active</td>
</tr>
<tr>
<td>SAML2</td>
<td>Inactive</td>
<td>SAML2_update1</td>
<td>Active</td>
</tr>
<tr>
<td>SAML2Logout</td>
<td>Inactive</td>
<td>SAML2Logout_update1</td>
<td>Active</td>
</tr>
</tbody>
</table>

You can navigate to the SAML 2.0 login and logout installation exit scripts using these paths.

- **SAML 2 Single Sign-on > Login script.**
- **SAML 2 Single Sign-on > Logout script.**
- **System Definition > Installation Exits.**

You can navigate to the SAML 2.0 update 1 script include using these paths.

- **SAML 2 Single Sign-on > Script object.**
- **System Definition > Script Includes.**

Test the Update

Perform these steps to troubleshoot your integration update.

1. **Add a new system property** called glide.authenticate.sso.saml2.debug with a value of true.
2. Attempt SAML 2.0 login.
3. Review the system log. SAML2 validation errors begin with the text SAML2ValidationError.
4. Identify and fix any common login errors.

Sample SAML 2 responses after the update
The following sections illustrate the new required elements and attributes that the IdP should provide in the SAML Response.

Sample SAML 2 Response with Issuer Element
The following SAML 2 response uses the Issuer element.

```xml
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
Destination="https://demoi2.service-now.com/navpage.do"
ID="s28da6774c88ae1eb292bf25fe625db81919d8e1e"
InResponseTo="SNC841720c227c81948cf68cadcdad235c6"
IssueInstant="2012-01-30T20:07:10Z" Version="2.0">
  <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    http://idp.ssocircle.com
  </saml:Issuer>
  ...
  <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
ID="s2f347f973c063836cf70eaa38302d94976f9c5b851"
IssueInstant="2012-01-30T20:07:10Z" Version="2.0">
    http://idp.ssocircle.com
  </saml:Assertion>
  ...
</samlp:Response>
```

Sample SAML 2 Response with the SubjectConfirmation and SubjectConfirmationData Elements
The following SAML 2 response uses the SubjectConfirmation and SubjectConfirmationData elements with the NotOnOrAfter and Recipient attributes.

```xml
<saml:SubjectConfirmationMethod="urn:oasis:names:tc:SAML:2.0:cm:bearer">
  <saml:SubjectConfirmation
InResponseTo="SNC841720c227c81948cf68cadcdad235c6"
NotOnOrAfter="2012-01-30T20:17:10Z"
Recipient="https://demoi2.service-now.com/navpage.do"/>
</saml:SubjectConfirmation>
```

Sample SAML 2 Response with the AudienceRestrictions and Audience Elements
The following SAML 2 response uses the AudienceRestrictions and Audience elements with the NotBefore and NotOnOrAfter attributes.

```xml
<saml:ConditionsNotBefore="2012-01-30T19:57:10Z"
NotOnOrAfter="2012-01-30T20:17:10Z">
  <saml:AudienceRestriction>
    https://demoi2.service-now.com
  </saml:AudienceRestriction>
</saml:Conditions>
```

Clone an instance with a SAML integration
Preserving SAML SSO-related settings can prevent the target instance from redirecting all authentication requests to the original IdP with the wrong issuer and audience parameters.

To preserve SAML settings, create data preservers for the following tables:
SAML user provisioning

If users exist in your IdP but are not in your instance, SAML user provisioning can automatically create the users in your instance's User [sys_user] table.

SAML user provisioning is supported for SAML 2.0 Update 1 when Multi-SSO is enabled.

How SAML user provisioning works

When SAML user provisioning is enabled and the system encounters a new user that is not in the instance, the instance automatically creates a record in a temporary table with the name \texttt{u\_import\_saml\_user\_<suffix>}, where \texttt{<suffix>} is an automatically generated text identifier. The system also creates transform map that specifies the data relationships between the import table and the User table. Each IdP in identified in the system has its own transform map. The transform map is created once for each IdP. Administrators can update it as necessary.

When the user logs in, they access an IdP to log in.

- The system presents a list of all IdPs that are able to use SAML user provisioning. If there is only one IdP that can use SAML user provisioning, that one is used automatically.
- If none of the above conditions are true, the system uses the default IdP.

Administer SAML user provisioning

To update the User table with the users in your IdP, you must first set up field mapping and then enable user provisioning through Multi-SSO IdP settings.

Set up your IdP mapping to identify what fields in the IdP are mapped to the correct fields in the User table.

Role required: admin

1. Navigate to Multi-Provider SSO > Properties.
2. Select Enable Auto Importing of users from all identity providers into the user table (glide.authenticate.multisso.user.autoprovision) to activate this feature.
3. Click Save.
5. Open the SAML2 Update1 record.
6. To create a record in the User table when the user does not already exist, select Auto-provision Users. If you upgraded to this release, you need to configure the form and add this field.
7. To enable user records to be updated when users log in to the IdP and the information on the IdP is out of date with the information on the User table, select Update User Record Upon Each Login. If you upgraded to this release, you need to configure the form and add this field.
8. Click User Provisioning Transform Map to see the map that the system automatically creates.
9. Make changes to the map as needed.

When the first unknown users tries to log in, the system creates the fields in the import set table from the metadata.xml file.

\textbf{Note:} You cannot map the fields from the IdP table until this first user logs in.
SAML 2.0 troubleshooting

Before contacting support, try the troubleshooting solutions available in the knowledge base on Hi.

**Note:** The instance does not support solutions provided by external sites.

See the following knowledge base article: [KB0540617 “SAML Error Matrix”](#).

### Table 462: Other Common Issues

<table>
<thead>
<tr>
<th>Error or Symptom</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error message: “is not a function.”</td>
<td>This error occurs because the plugin was not active and did not load the .jar file. Therefore, the code appears to be missing. Contact Technical Support to restart nodes that are missing the plugin.</td>
</tr>
<tr>
<td>This issue might occur in a multi-node environment. If the plugin does not get activated on all nodes, an error like the following appears: org.mozilla.javascript.EcmaError: [JavaPackage org.opensaml.saml2.core.impl.AuthnRequestBuilder] is not a function.</td>
<td></td>
</tr>
<tr>
<td>SAML does not authenticate users accessing CMS pages.</td>
<td>By default, CMS pages are public and therefore do not require authentication. If you want SAML to authenticate CMS pages, change the view_content.do public page from active=true to active=false.</td>
</tr>
<tr>
<td>Cannot redirect a user back to a CMS page after SAML authentication.</td>
<td>By default, the SSO integration uses a URL parameter called URI to control where the user is directed after authentication at the IdP. SSO ignores relative URLs. For example, SSO cannot redirect users to a /ess relative URL. Instead, the user has to navigate to a URL such as /nav_to.do?uri=/ess, which uses deep linking syntax. However, this puts the ESS portal inside the main navigation content IFrame. In other words, the site does not take up the full page, but rather loads as a page in your instance. For more information, see CMS Sites and Single Sign-On. If you change the CMS entry page to make it private by setting view_content.do to active=false, deep linking behavior then requires a customization to the Installation Exit login script. Create a script that looks for the URI portion of the URL and constructs a RelayState URL parameter containing the relative URL path to redirect users after authenticating at the IdP.</td>
</tr>
</tbody>
</table>
Geneva  ServiceNow  ServiceNow Platform

<table>
<thead>
<tr>
<th>Error or Symptom</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML does not redirect users to the appropriate page after authentication.</td>
<td>Determine if the relay state is passed out to the IdP and then passed back during authentication. You can do this with a browser capable of saving HTTP request headers and POST info, such as Chrome with its built-in developer tools, or Firefox with the add-on called HTTPfox. For Internet Explorer, use a third-party application such as Fiddler. The goal is to watch the requests pass from the client (browser) to the instance, and from the client to the IdP.</td>
</tr>
</tbody>
</table>

Monitor the event queue for login activities

Every single sign-on integration creates events for login activities.

You can use these events to monitor for login failures and determine if there are any security concerns to address.

Table 463: Monitoring the event queue for login failures

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Description</th>
<th>Record</th>
<th>Parameter 1</th>
<th>Parameter 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>external.authentication.succeeded</td>
<td>External authentication succeeded and the user accessed the instance URL.</td>
<td>Session ID</td>
<td>User ID of user who successfully logged in</td>
<td>The URL the user accessed (which may be a deep link)</td>
</tr>
<tr>
<td>external.authentication.failed</td>
<td>The single sign-on requirements are not present or are missing.</td>
<td>Session ID</td>
<td></td>
<td>The missing authentication requirements</td>
</tr>
<tr>
<td>external.authentication.failed</td>
<td>The user does not exist in the User [sys_user] table</td>
<td>User ID</td>
<td></td>
<td>The string, &quot;User does not exist&quot;</td>
</tr>
<tr>
<td>external.authentication.failed</td>
<td>The user is locked out.</td>
<td>User ID</td>
<td></td>
<td>The string, &quot;User locked out.&quot;</td>
</tr>
</tbody>
</table>

Event queue login events

The SAML 2.0 integration creates events for login activities.

You can use these events to monitor for login failures and determine if there are any security concerns to address.
### Table 464: Login activities events

<table>
<thead>
<tr>
<th>Event name</th>
<th>Description</th>
<th>Session ID</th>
<th>User ID</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>saml2.logout.validation.failed</td>
<td>The logout response from the IdP failed validation against your logout request. The event validates the &lt;inResponseTo&gt; element against the session ID (ID attribute of the <a href="">saml2p:LogoutRequest</a> element). For example, see the workflow for logout request issued.</td>
<td></td>
<td>Session ID</td>
<td>The string, &quot;SAML2 LogoutResponse validation failed.'</td>
</tr>
<tr>
<td>external.authentication.succeeded</td>
<td>External authentication succeeded and the user accessed the instance URL.</td>
<td></td>
<td>User ID of user who successfully logged in</td>
<td>The URL the user accessed (which may be a deep link)</td>
</tr>
<tr>
<td>external.authentication.failed</td>
<td>The single sign-on requirements are not present or are missing.</td>
<td></td>
<td>Session ID</td>
<td>The missing authentication requirements</td>
</tr>
<tr>
<td>external.authentication.failed</td>
<td>The user does not exist in the User [sys_user] table.</td>
<td></td>
<td>User ID</td>
<td>The string, &quot;User does not exist&quot;</td>
</tr>
<tr>
<td>external.authentication.failed</td>
<td>The user is locked out.</td>
<td></td>
<td>User ID</td>
<td>The string, &quot;User locked out.&quot;</td>
</tr>
</tbody>
</table>

## OKTA SSO integration

Administrators can enable the Okta SSO integration to provide single sign-on access to instances through Okta.

This integration allows users to access their instance without logging in with their user credentials. Users only need to sign on to their organization's Okta service to gain access to the instance.

### How the Okta integration works

Organizations using Okta for single sign-on services typically have user information stored in another user management system, such as Active Directory. The user data is shared between Okta and these services or applications so that Okta always has the most up-to-date user credentials necessary to automatically log users into their applications. Organizations using Okta might also enter user data directly into the Okta user management service.
For Okta to automatically authenticate users to the instance, user data must be shared between Okta and the instance. The user management system in Okta manages this data sharing. In the user management system, administrators identify the users that are users by either manually configuring the user data or by importing users from the database. If changes need to be made at a later time, the user management system can import the updates from the database or, if the updates were made in Okta, the user management system can push the updates to the user table. This keeps the user data in synch.

When a user successfully authenticates with Okta and tries to access the instance, a token is sent to the instance authorizing the user so they do not have to sign in again.

Activate OKTA SSO

An administrator can activate the SSO Provided by Okta, Inc. plugin, which also activates the SAML 2.0 Update 1 plugin.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Set up the OKTA SSO integration

The OKTA site contains the up-to-date instructions on how to set up the integration with your instance.

Role required: admin

1. Activate the SSO Provided by Okta, Inc. plugin.
2. Navigate to the following external URL: ServiceNow Deployment Guide page.
3. Download the documentation.

Instruct users to log in with OKTA SSO

Users should log in to Okta before logging in to the application.

Role required: admin

This gives them access to all the applications that are configured to work with Okta, including the application, without signing in to each one separately.

After you activate Okta SSO, users can log into the application by:

- Clicking the Click here to log in with Okta, Inc. link on the login page.
- Clicking the application link on your organization’s Okta site.

In both cases, users who are already logged in to Okta are immediately redirected to the instance as a logged-in user. If they are not logged in to Okta, they must log in with their Okta credentials before being automatically redirected to the instance.
Digest token authentication

The digest token authentication passes user credentials and a digest token within an unencrypted HTTP header.

The instance reads the HTTP header value and compares its computed hash value of the digest token. If the computed hash value matches the digest token value, then the instance searches for a matching value in the User table. If there is a matching value in the User table, the instance considers the user pre-authenticated and logs the user in.

Digest token authentication is more secure than simple unencrypted HTTP headers because any accidental or intentional change to the unencrypted HTTP header produces a different hash value. If the hash value fails to match, the instance denies the user access to the requested instance. This prevents users from attempting to login with another user’s credentials.

Integration requirements

A Digest Token Authentication integration requires:

- A web server
- SiteMinder or another single sign-on application to pre-authenticate the user on the local network
- A web page or portal that passes user credentials to the target instance in one of these formats
  - HTTP Header
  - URL parameter
  - Cookie
- A web page or portal that creates and passes a digest token to the target instance using one of these encoding techniques
  - SHA1
  - MD5

Set up Digest Token Authentication

Setting up digest token authentication involves generating an unencrypted HTTP header and token, then enabling an installation exist and configuring properties.

Generate an unencrypted HTTP header

The first step in setting up digest token authentication is to generate an unencrypted HTTP header.

Role required: admin

The system only accepts one HTTP header as the source of pre-authentication.

1. Choose an HTTP header that matches data from the User (sys_user) table.
2. Configure your single sign-on software or authentication portal to generate the desired HTTP header.

For example, here are some sample SiteMinder HTTP headers.

<table>
<thead>
<tr>
<th>User credential</th>
<th>Sample HTTP header</th>
<th>Matching sys_user field</th>
</tr>
</thead>
<tbody>
<tr>
<td>User name</td>
<td>SM_USER</td>
<td>user_name</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
User credential | Sample HTTP header | Matching sys_user field
--- | --- | ---
Email address | EMAIL | email

Generate a digest token
After generating an unencrypted HTTP header, generate a digest token for authentication.

Role required: admin

The system only accepts one HTTP header as the source of pre-authentication.

Configure your single sign-on software or authentication portal to generate an HTTP header to contain the digest token.

For example, here is a sample SiteMinder HTTP header.

Table 466: Sample SiteMinder HTTP headers

<table>
<thead>
<tr>
<th>Sample HTTP header</th>
<th>Sample digest HTTP header</th>
<th>Sample HTTP headers with digest</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM_User</td>
<td>DE_USER</td>
<td>SM_USER=joe.employee;DE_USER=KfNr4nlt86b9gEfcoiT0dU/YLM0=</td>
</tr>
</tbody>
</table>

Typically, the authentication portal or web page runs the code to generate a digest token from the unencrypted HTTP header and includes the digest token in the URL to the target instance. See SNC Guru for code samples of generating a digest token.

Enable digest installation exit script
After generating a digest token for authentication, enable a digest installation exist script.

Role required: admin

The system uses an installation exit to identify and decode incoming digest tokens. By default, the installation exit uses SHA1 to compute a hash value.

1. Navigate to System Definition > Installation Exits.
2. Click DigestSingleSignOn.
3. Set Active to true.
4. Click Update.
5. To use MD5 for computing the digest token hash value, change this line: var MAC_ALG ="HmacSHA1"; to this: var MAC_ALG ="HmacMD5";

Enter single sign-on system properties
After enabling a digest installation exist script, configure properties for single sign-on. If you are configuring digest token for multi-provider SSO, skip this step and enter the digest properties for multi-provider single sign-on.

Role required: admin

Set these properties to enable single sign-on with an unencrypted HTTP header.

1. Navigate to System Properties > Single Sign-on.
2. For Enable external authentication, select Yes.
3. For **HTTP header name to look for an externally authorized user (e.g. for SiteMinder)**, enter the HTTP header you generated. For example, `SM_USER`.

4. For **HTTP header name to look for an externally authorized user (e.g. for Digest Encryption)**, enter the HTTP header you generated. For example, `DE_USER`.

5. For **Service-now.com field name to match against the incoming header**, enter the `sys_user` field that contains matching data. For example, `user_name`.

6. For **Secret passphrase for single sign-on (SSO) encryption/decryption**, enter the secret key (password) to use for encoding digest keys. For example, `SecretKey123`.

7. Click **Save**.

---

**Enter the digest properties for multi-provider single sign-on**

After enabling a digest installation exist script, configure properties for multi-prover single sign-on. If you are not using multi-provider single sign-on, configure standard single sign-on properties.

**Role required:** admin

1. In the **Name** field, enter the name of the digest token.
2. In the **User** field, enter the `sys_user` field that contains the matching data for the incoming header.
3. In the **HTTP Digest header name** field, enter the HTTP header you generated. For example, `DE_USER`.
4. In the **HTTP header name field**, enter the HTTP header you generated in Step 1. For example, `SM_USER`.
5. In the **Secret Passphrase** field, enter the secret key to use for encoding digest keys. For example, `SecretKey123`.
6. In the **Failed SSO Redirect field**, enter the URL to redirect users after a failed authentication.
7. In the **External logout redirect** field, enter the URL to redirect users after a logout.
8. In the **Single Sign-on Script**, select `MultiSSO_DigestedToken`.
9. Click **Update**.
Test the integration
This single sign-on method allows pre-authentication from URL parameters, HTTP headers, or cookies.

Role required: admin

Verify users can sign-on from all methods.

For example:

- **URL Format**: https://<instance name>.service-now.com?
  SM_USER=joe.employee&DE_USER=KfNr4nlt86b9gEfoiT0dU/YLM0=

- **HTTP Header Format**:
  SM_USER=joe.employee;DE_USER=KfNr4nlt86b9gEfoiT0dU/YLM0=

- **Cookie Format**: 'SM_USER=joe.employee;
  DE_USER=KfNr4nlt86b9gEfoiT0dU/YLM0=; expires=Fri, 27 Jul 2012
  02:47:11 UTC; path=/'

Create links for digest authentication
After you set up digest token authentication, construct URLs to take users to a particular page or resource.

Role required: admin

- Create a URL to your instance and append the appropriate user (SM_USER) and digest (DE_USER) request parameters.

Example URLs are as follows:

  https://demo.service-now.com/nav_to.do?
  uri=ess&SM_USER=itil&DE_USER=nqQ82LONeYQnXPmdxnsvsArrQ=

  https://demo.service-now.com/nav_to.do?uri=incident.do?
  sys_id=9d385017c611228701d22104cc95c371&SM_USER=itil&DE_USER=Uj/
  HopjjPczCNpN2xcCX17kQty4=

  http://demo.service-now.com/nav_to.do?uri=incident_list.do?
  sysparm_userpref_module=4dedbac7d0bbf538017a7f13c4af257%26sysparm_query=caller_id=javascript:gs.getUserID()^active=true^EQ&SM_USER=itil&DE_USER=Uj/
  HopjjPczCNpN2xcCX17kQty4=

**Note**: If you are using request parameters inside a URI (nav_to.do?uri=), ampersands (&) are encoded as %26.

Sample digest token implementations
Here are several samples of creating a digest token.

**Table 467: Sample digest authentication implementations**

<table>
<thead>
<tr>
<th>Digest built with</th>
<th>Secret key value</th>
<th>Hash method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>abc123</td>
<td>SHA1</td>
<td>Sample Java Digest Algorithm for Encryption</td>
</tr>
</tbody>
</table>
Sample Java digest algorithm for encryption

This Java algorithm illustrates creating a digest token from an HTTP header.

This sample assumes:

- The web server supports Java
- The hash computation method is SHA1
- The secret key value is abc123
- The unencrypted HTTP header name is user_name

Change the Java code to use another hash computation mechanism (such as MD5), change the secret key value, or HTTP header name.

```java
import javax.crypto.Mac; import javax.crypto.spec.SecretKeySpec; import sun.misc.BASE64Encoder; // public class DigestTest { private static final String MAC_ALG = "HmacSHA1"; // default to something JDK 1.4 has String fKey = "abc123"; public byte[] getDigest ( String acct ) { try { byte[] bkey = fKey.getBytes( ); byte[] data = acct.getBytes( ); Mac mac = null; try { mac = Mac.getInstance (MAC_ALG ); mac. init ( new SecretKeySpec (bkey, MAC_ALG ) ) ; } catch ( Exception e ) { e.printStackTrace ( ) ; } byte[] sig = mac. doFinal (data ) ; String signature = new String (sig ) ; System. out. println ( "value:" + acct ) ; System. out. println ( "digested value:  " + signature ) ; return sig ; } catch ( IllegalStateException e ) { e.printStackTrace ( ) ; } return null ; } public static void main ( String[] args ) { BASE64Encoder encoder = new BASE64Encoder ( ) ; DigestTest test = new DigestTest ( ) ; String userName = "user_name" ; System. out. println ( "base 64 digest username: " + encoder. encode (test. getDigest (userName ) ) ) ; } }
```

Sample C

This C class illustrates creating a digest token from three input parameters.

- strEncryptionMethod – lists the hash computation method (SHA1 or MD5)
- message – lists the value to be converted into a digest token
- sharedKey – lists the secret key

This sample assumes:

- The web server supports C
- Other code calls this class and passes the expected parameters
Sample Code

```csharp
private string digestData(string strEncryptionMethod, string message, string sharedKey) {
    UnicodeEncoding myUnicodeEncoding = new UnicodeEncoding();
    byte[] messageBytes = System.Text.Encoding.ASCII.GetBytes(message);
    byte[] sharedKeyBytes = System.Text.Encoding.ASCII.GetBytes(sharedKey);
    byte[] hashedMessage;
    string b64SHA1Message;

    if (this.DEBUG) {
        MessageBox.Show(message);
        TextBoxSecret.Text = sharedKey;
    }

    switch ((strEncryptionMethod))
    {
        case "SHA1":
            HMACSHA1 hmacsha1 = new HMACSHA1();
            hmacsha1.Key = sharedKeyBytes;
            hashedMessage = hmacsha1.ComputeHash(messageBytes);
            b64SHA1Message = Convert.ToBase64String(hashedMessage);
            if (this.DEBUG) MessageBox.Show(Convert.ToString(hashedMessage));
            break;
        case "MD5":
            HMACMD5 hmacmd5 = new HMACMD5(sharedKeyBytes);
            hashedMessage = hmacmd5.ComputeHash(messageBytes);
            b64SHA1Message = Convert.ToBase64String(hashedMessage);
            if (this.DEBUG) MessageBox.Show(Convert.ToString(hashedMessage));
            break;
        default:
            b64SHA1Message = "Unknown Encryption Method";
            break;
    }

    MessageBox.Show(b64SHA1Message);
    return b64SHA1Message;
}
```

OAuth 2.0

Your instance supports Open Authorization (OAuth) 2.0.

Configure OAuth 2.0 settings for these scenarios:

- **OAuth external client scenario**: Your instance provides an endpoint for third-party clients to pull data from the instance.
- **OAuth provider scenario**: Your instance pulls data from a third-party provider.
Attention: Using OAuth to allow your users to authenticate to the instance is not currently supported.

Both the simple security and high security frameworks support OAuth 2.0. High Security is recommended. See High Security Settings on page 2437 for information about which versions have high security already active and how to activate high security. You must have the security_admin role to manage the OAuth integration.

**OAuth 2.0 concepts**

The following table describes the key concepts of the OAuth 2.0 implementation.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Owner</td>
<td>An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user. The resource owner is always a user account.</td>
</tr>
<tr>
<td>Client</td>
<td>An application making protected resource requests on behalf of the resource owner and with its authorization.</td>
</tr>
<tr>
<td>Resource Server</td>
<td>The server hosting the protected resources, capable of accepting and responding to protected resource requests.</td>
</tr>
<tr>
<td>Authorization Server</td>
<td>The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.</td>
</tr>
<tr>
<td>Authorization Request</td>
<td>The permission a client needs to access a protected resource. The authorization request is always an HTTP POST message containing the ID of the client acting on the resource owner's behalf and credentials authorizing the request.</td>
</tr>
<tr>
<td>Authorization Grant</td>
<td>A credential representing the resource owner's authorization to access a resource. The authorization grant is either a user's login credentials or a refresh token.</td>
</tr>
<tr>
<td>Access Token</td>
<td>A secure string that a client uses to access protected resources. An instance issues access tokens to clients that have a valid authorization grant. Each access token has a specific scope, lifespan, and other attributes. By default, an instance issues access tokens with a 30 minute lifespan in the scenario where the instance is the OAuth provider. For third-party tokens, 30 days.</td>
</tr>
<tr>
<td>Concept</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Refresh Token</td>
<td>A credential that a client uses to obtain new access tokens without requiring additional user authorization. An instance issues a refresh token to a client when it is first authorized to have an access token. By default, an instance issues refresh tokens with a 100 day lifespan in the scenario where the instance is the OAuth provider. For third-party tokens, 365 days.</td>
</tr>
</tbody>
</table>

**OAuth process**

In general, an instance uses the following workflow to authorize access to an OAuth-protected resource.

1. The client requests authorization from the resource owner through the authorization server and the resource owner's login credentials or an authorization code.
2. The authorization server provides the client with an authorization grant on behalf of the resource owner.
3. The client requests an access token from the authorization server using the authorization grant. If valid, the authorization server grants the client an access token and refresh token.
4. The client uses the access token to request the protected resource.
5. The resource server validates the access token.
6. If valid, the resource server grants the client access to the protected resource.

**OAuth grant type**

*Grant type* refers to

- **Authorization code**: the consumer sends an authorization code to get an access token. You can specify this by creating an *OAuth profile*. This is also referred to as OAuth code flow.
- **Resource owner password credentials**: the consumer of the resource already has the user credentials to get the access token. This is also referred to as password flow.
  
  The OAuth application supports both grant types.

**Storage of authentication credentials**

The OAuth client secret is stored as a *password2* type field, which is encrypted in Triple DES. User passwords, which are used to check incoming endpoint requests, are stored as a hash value in the User table in a *password* type field (SHA 256).

**OAuth enhancements**

Brief introduction to the topic that appears on overview pages and in link previews.
OAuth third-party provider application registries now use profiles and scopes:

- The OAuth profile specifies the grant type (authorization code flow or password flow) and the scope. See Specify an OAuth profile on page 1962 for more information.
- OAuth scopes specify the level of access to a restricted resource. Typically access is write or read. See Specify an OAuth scope on page 1964 for more information.

- The getToken method in GlideOAuthClient allows you to retrieve the access token, refresh token, and expiration date and time for the access token.
- OAuth supports the authorization code flow grant type, which you can specify in an OAuth profile. This feature allows a user to access a resource by authenticating directly with a third-party OAuth server that trusts the resource, instead of authenticating with username + password credentials.
- OAuth supports application scoping by providing access by providing access to all OAuth methods using the sn_auth prefix. See Scoped OAuth APIs on page 1968.
- A new method, requestTokenByRequest, is available for the GlideOAuthClient class.

Activate OAuth

By default, the OAuth plugin is active on new and upgraded instances. If it is not active on your instance, you can activate it.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Set up OAuth

Set up OAuth by registering applications to OAuth, enable the OAuth system property, and generate token requests.

Role required: admin

You must have the security_admin role to set up OAuth.

1. Activate the plugin.
2. Register the application to be authorized by OAuth. You can do this for:
   
   - External clients
   - Third-party OAuth providers
3. Set the OAuth property.
4. Configure applications to create an HTTP POST requesting an OAuth token. The application must also be able to parse the JSON response to use the provided access and refresh tokens.

Set the OAuth property

You must add a property to generate OAuth 2.0 tokens to registered applications.
Role required: admin

1. Type `sys_properties.list` in the application navigator filter.
2. Click New.
3. Fill out the form with the following settings:
   - **Name**: com.snc.platform.security.oauth.is.active
   - **Type**: true | false
   - **Default value**: false
4. Set the property to **true** to use OAuth 2.0.

*Create endpoint for external clients*

Each client application must register with the instance to participate in OAuth authorization.

Role required: admin

1. Navigate to **System OAuth > Application Registry**.
2. Click **New**.
3. On the interceptor page, click Create an OAuth API endpoint for external clients.
4. Fill out the fields, as appropriate (see table).
5. Click **Submit**. The record is saved in the Application Registries [oauth_entity] table.
Table 469: Application registration field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name identifying the application you are requiring OAuth access for.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Client ID</td>
<td>[Read-Only] The automatically-generated unique ID of the application. The instance uses the client ID when requesting an access token.</td>
</tr>
<tr>
<td>Client Secret</td>
<td>[Required] The shared secret string that both the instance and the application use to authorize communications with one another. The instance uses the client secret when requesting an access token. Leave this field blank to have the instance automatically generate a client secret.</td>
</tr>
<tr>
<td>Logo URL</td>
<td>The URL containing an image to use as the application logo.</td>
</tr>
<tr>
<td>Active</td>
<td>A check box for indicating that the instance can authorize access to the application. Only active applications can request access tokens.</td>
</tr>
<tr>
<td>Refresh Token Lifespan</td>
<td>The number of seconds a refresh token is valid. The instance uses this lifespan value when requesting a refresh token. By default refresh tokens expire in 100 days (8640000 seconds).</td>
</tr>
<tr>
<td>Access Token Lifespan</td>
<td>The number of seconds an access token is valid. The instance uses this lifespan value when requesting an access token. By default access tokens expire in 30 minutes (1800 seconds).</td>
</tr>
<tr>
<td>Comments</td>
<td>Any additional information you want to associate with this application.</td>
</tr>
</tbody>
</table>

*Use a third-party OAuth provider*

Each client application must register with the instance to participate in OAuth authorization.

Role required: admin

1. Navigate to **System OAuth > Application Registry**.
2. Click **New**.
3. On the interceptor page, click **Connect to a third party OAuth Provider**.
4. Fill out the fields, as appropriate (see table).
5. Click **Submit**. The record is saved in the Application Registries [oauth_entity] table.
OAuth provider details:
- Name: New Test
- Client ID: client076237
- Client Secret: Password
- OAuth MPI Social: Unconfigured
- Token URL: http://datatables.com
- Redirect URL: http://channel/users.service
- Refresh Token Lifespan: 0,040,300
- Accessible from: All application scopes

OAuth Entity Profiles:

Define the OAuth Provider Profile including name, Grant Type and OAuth Scope combinations. Click each profile entry to update the OAuth scopes. When a new OAuth provider is created, the system automatically generates a default profile for it without any scope.

OAuth Entity Profiles:

- Name: New Test
- Is default: No
- Grant type: None
- Authorization Code:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name identifying the application you are requiring OAuth access for.</td>
</tr>
<tr>
<td>Client ID</td>
<td>The unique ID of the application. The instance uses the client ID when requesting an access token.</td>
</tr>
<tr>
<td>Client Secret</td>
<td>[Required] The shared secret string the instance and the application use to authorize communications with one another. The instance uses the client secret when requesting an access token. Enter a string.</td>
</tr>
<tr>
<td>OAuth API Script</td>
<td>The script used to customize the request and response to the third-party OAuth provider. The script name must have the prefix <code>OAuth</code>.</td>
</tr>
<tr>
<td>Logo URL</td>
<td>The URL containing an image to use as the application logo.</td>
</tr>
<tr>
<td>Default Grant Type</td>
<td>The type of grant:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Authorization code</strong>: The code granted to the client to obtain an access token, which is then used to obtain access to the resource. You need an authorization URL if you select this option.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Resource owner password credentials</strong>: The user name and password of the user trying to obtain access to the resource.</td>
</tr>
<tr>
<td>Refresh Token Lifespan</td>
<td>The refresh token lifespan in seconds.</td>
</tr>
<tr>
<td>Accessible from</td>
<td>The application scope that this registry is accessible from.</td>
</tr>
<tr>
<td>Active</td>
<td>A check box for indicating that the instance can authorize access to the application. Only active applications can request access tokens.</td>
</tr>
<tr>
<td>Authorization URL</td>
<td>The URL of the endpoint to authorize the user if you are using the authorization code grant type.</td>
</tr>
<tr>
<td>Token URL</td>
<td>The location of the token endpoint that the instance uses to retrieve and refresh tokens.</td>
</tr>
<tr>
<td>Redirect URL</td>
<td>The application endpoint that receives the authorization code. Leave this field empty to have the instance automatically generate this URL.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token Revocation URL</td>
<td>The location of the endpoint that the instance uses to revoke the token.</td>
</tr>
<tr>
<td>Comments</td>
<td>Any additional information you want to associate with this application.</td>
</tr>
</tbody>
</table>

**Embedded lists**

- **OAuth Entity Profiles**
  - The profiles associated with the OAuth provider. The profile includes the grant type. Click the profile name to go to the OAuth Entity Profile form.

- **OAuth Entity Scopes**
  - The entity scopes associated with the OAuth provider. The scope identifies the services the application has access to. Click the scope name to go to the OAuth Entity Scope form.

---

After you create the third-party application registry, the system automatically generates a default profile using the specified grant type, but without any scopes. You can create additional profiles, each with scopes.

**OAuth support for authorization code flow**

OAuth 2.0 authorization code flow allows a user to access a resource by authenticating directly with a third-party OAuth server that trusts the resource, instead of authenticating with username/password credentials.

This implementation of OAuth authorization code flow allows a user to get access to a resource via REST. The authorization code framework automatically gets the access and refresh tokens through the authorized URL that the administrator configures, instead of requiring the user to enter a user name and password to get access to the resource.

**OAuth profiles and scopes**

In the OAuth provider scenario, profiles and scopes specify the grant type, authorization type, and the level of access.

In the OAuth provider scenario, the OAuth profile refers to a combination of a grant type and at least one scope. The scope specifies the access that the user has to the protected resource, such as read or write. You can create a profile for each third-party provider and obtain the specific set of scopes from the provider. See Specify an OAuth profile on page 1962 and Specify an OAuth scope on page 1964 for more information. The instance also uses OAuth profiles when a REST call specifies OAuth 2.0 authentication. A default profile is automatically created for each third-party provider record that you create. There can only be one default.

OAuth profiles and scopes are available with the Geneva release.

Specify these parameters, which are saved in the OAuth Requestor Profile [OAuth_requestor_profile] table:

**Table 471: OAuth parameters for default profile support**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oauth_requestor</td>
<td>The Sys ID of the object, which can be a user record or an email account.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oauth_requestor_context</td>
<td>A descriptor that provides context for the oauth requestor. As a good practice, use the name of the table where the <code>oauth_requestor</code> object is saved.</td>
</tr>
<tr>
<td>oauth_provider_profile</td>
<td>The Sys ID of the OAuth profile record that is the default.</td>
</tr>
</tbody>
</table>

When the user attempts to authenticate, the provider accesses the OAuth Requestor Profile table to look for the user. If the user is found, the authentication is successful. If not, the provider accesses the default profile to determine the grant type and how to proceed with the authentication.

Specify an OAuth profile
An OAuth profile includes the grant type that the third-party OAuth provider needs to obtain access to the restricted resource.

Role required: admin

1. Open a third-party OAuth provider record.
2. In the OAuth Entity Profiles embedded list, click **Insert a new row**.
3. Enter a name for the profile.
4. Right-click Application Registry form header and select **Save**.
   The profile record is created.
5. Click the name of the profile you created.
6. Fill in the form fields (see table).
### Table 472: OAuth Entity Profile form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>OAuth provider</td>
<td>Verify the provider associated with this profile.</td>
</tr>
<tr>
<td>Grant type</td>
<td>Select the grant type:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Authorization code</strong>: The code granted to the client to obtain an access token, which is then used to obtain access to the resource. You need an authorization URL if you select this option.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Resource owner password credentials</strong>: The user name and password of the user trying to obtain access to the resource.</td>
</tr>
</tbody>
</table>
7. Click **Update**.

Specify an OAuth scope

Specify the OAuth scopes that you get from the provider. Scopes can be any level of access specified by the provider, such as read, write, or any string, including a URL.

Role required: admin

1. Open a third-party OAuth provider record.
2. Open a profile associated with the provider.
3. In the OAuth Entity Profile Scopes embedded list, click **Insert a new row**.
4. Enter a name for the profile.
5. Right-click OAuth Entity Profile form header and select **Save**.
   The profile record is created.
6. Click the name of the scope you created.
7. Fill in the form fields (see table).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is default</td>
<td>Select this option to make this profile the default option for the associated provider.</td>
</tr>
<tr>
<td>Embedded list</td>
<td>OAuth Entity Profile Scopes *Specify the OAuth entity scope.*</td>
</tr>
</tbody>
</table>

Table 473: OAuth Entity Scope form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>OAuth provider</td>
<td>Verify the provider associated with this scope.</td>
</tr>
<tr>
<td>OAuth scope</td>
<td>The scope that you are granted by the provider. Typical scopes are read and write. Scopes can be any string that the provider specifies.</td>
</tr>
</tbody>
</table>
8. Click Update.

**OAuth tokens**
Instances offer an OAuth 2.0 token generation API for client applications to access instance data on behalf of a user account. The API accepts HTTP POST messages to the access token endpoint URL and returns a JSON response containing the access token.

**Access token request format**

You can use the OAuth 2.0 API to construct an access token request.

- **Endpoint URL**: `<instance-name>.service-now.com/oauth_token.do`
- **Request format**: All requests should formatted as a URL-encoded HTTP POST body.
- **Request parameters**: Most request parameters are required.
- **Response format**: JSON
- **Response parameters**: The response contains one or more OAuth tokens.

**OAuth parameters for default profile support**

The default profile feature require a set of parameters that you can use with the setParameter() API to specify the oauth requestor, a context for the request, and the provider profile.

In the OAuth provider scenario, you must set three parameters that tells the OAuth provider which OAuth profile to use by default. When these three parameters are set, the access token is saved in the instance database. Use these parameters with GlideOAuthClientRequest.

These parameters are available starting with the Geneva release.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oauth_requestor</td>
<td>The Sys ID of the object, which can be a user record or an email account.</td>
</tr>
<tr>
<td>oauth_requestor_context</td>
<td>A descriptor that provides context for the oauth requestor. As a good practice, use the name of the table where the oauth_requestor object is saved.</td>
</tr>
<tr>
<td>oauth_provider_profile</td>
<td>The Sys ID of the OAuth profile record that is the default.</td>
</tr>
</tbody>
</table>

You do not need to set the grant type and scope using parameters. These items are configured in the OAuth profile record. If you do not use these parameters, you can use the setScope and setGrantType methods. See the GlideOAuthClientRequest setScope method and setGrantType methods. Access the API on the developer portal at https://developer.servicenow.com.

**OAuth API request parameters**

Access token requests use the following request parameters.

See the ServiceNow Developers site for API information.
### Table 475: Access token request parameters

<table>
<thead>
<tr>
<th>Request parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **grant_type**    | [Required] The type of credentials authorizing the request for an access token. This parameter must have a value of either password or refresh_token:  
  - **password**: A set of user credentials authorize the access token request. Specify the user credentials in the username and password parameters.  
  - **refresh_token**: An existing refresh token authorizes the access token request. Specify the refresh token in the refresh_token parameter. |
| **client_id**     | [Required] The automatically-generated unique ID of the client application requesting the access token. |
| **client_secret** | [Required] The shared secret string the instance and the OAuth application use to authorize communications with one another. |
| **username**      | The user account name authorizing the access token request. This parameter is required for access token requests with a **grant_type** of password. |
| **password**      | The password for the user account authorizing the access token request. This parameter is required for access token requests with a **grant_type** of password. |
| **refresh_token** | The existing refresh token authorizing the access token request. This parameter is required for access token requests with a **grant_type** of refresh_token. |

**Requests Using User Credentials**

The instance requires clients to provide user login credentials when first authorizing the client or when authorizing the creation a new refresh token. This type of request always produces two tokens in the response:

- An access token
- A refresh token

The instance verifies that the user is active, not currently locked out, and has an interactive session. If any of these conditions are false, the instance does not produce an access token. Access requests made within the access token's expiration time always return the current access token.
Note: This type of authorization grant relies on TLS encryption to protect the user's credentials during transmission.

The following example illustrates requesting an access token with a set of user credentials. Spaces have been added to improve readability.

```
$ curl -d"grant_type=password&client_id=be3aeb583ace210011c15b24a43e25d8
&client_secret=client_password
&username=admin&password=admin"
https://instancename.service-now.com/oauth_token.do
```

Requests Using a Refresh Token

The instance can use an existing refresh token to create a new access token. This type of request only produces an access token in the response. The instance confirms that the refresh token has not expired before generating a new access token. Access requests made within the refresh token's expiration time always return the current refresh token. Transmitting refresh tokens is generally more secure than transmitting user credentials. The following example illustrates requesting an access token with an existing refresh token. Spaces have been added to improve readability.

```
$ curl -
-d"grant_type=refresh_token&client_id=be3aeb583ace210011c15b24a43e25d8
&client_secret=client_password
&refresh_token=w599voG89897rGVDmdp12WA681r9E5948c1CJTPi8g4HGC4NWaz62k6k1K0PMxHW4O84H8yOO3H
https://instancename.service-now.com/oauth_token.do
```

OAuth API response parameters

The OAuth 2.0 API produces a JSON response containing the following parameters as name:value pairs.

See the ServiceNow Developers site for API information.

Table 476: Access token response parameters

<table>
<thead>
<tr>
<th>Response parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scope</td>
<td>The amount of access granted by the access token. The scope is always useraccount, meaning that the access token has the same rights as the user account that authorized it. For example, if Abel Tuter authorizes an application by providing his login credentials, then the resulting access token grants the token bearer the same access privileges as Abel Tuter.</td>
</tr>
<tr>
<td>token_type</td>
<td>The type of token issued by the request as defined in the OAuth RFC. The token type is always Bearer, meaning that anyone in possession of the access token can access a protected resource without providing a cryptographic key. See RFC6750 for more information about how OAuth 2.0 uses bearer tokens.</td>
</tr>
<tr>
<td>expires_in</td>
<td>The lifespan of the access token in seconds.</td>
</tr>
<tr>
<td>Response parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>refresh_token</td>
<td>The string value of the refresh token.</td>
</tr>
<tr>
<td>access_token</td>
<td>The string value of the access token. Access requests made within the access token's expiration time always return the current access token.</td>
</tr>
<tr>
<td>format</td>
<td>[Optional] The output format of the response. This value is always JSON.</td>
</tr>
</tbody>
</table>

The following example illustrates the JSON string returned by an access token request. Spaces have been added to improve readability.

```
{"scope":"useraccount","token_type":"Bearer","expires_in":1800,
"refresh_token":"w599voG89897rGVDmdp12WA681r9E5948c1CJTPi8g4HGC4NWaz62k6k1K0FMxHW4c",
"access_token":"F0jh9korTyzd9kaZqZ0SzjKZuS3ut0i4P46Lc52m2JYHiLcquistFAumpyxshU9mMQ13g
```

**Revoke an OAuth token**

You might want to revoke an OAuth access or refresh token for security reasons.

Role required: admin

Revoking the token pertains to the situation where your instance acts as the OAuth resource server.

- Access your instance using `oauth_revoke_token.do` and append the access or refresh token.
  
  For example: `https://[Your_ServiceNow_Instance]:[port]/oauth_revoke_token.do?token=[access or refresh token]` without the brackets `[ ]`.

This endpoint access does not require authentication. The token in this request is marked as expired.

**OAuth client APIs**

The OAuth client API provides methods to request and revoke OAuth tokens.

All APIs are available on the developer portal. Go to [https://developer.servicenow.com](https://developer.servicenow.com) to access the APIs. Search for the following APIs:

- GlideOAuthClient: Use these methods for requesting and revoking the refresh and access tokens.
- GlideOAuthClientRequest: Use these methods for handling client requests.
- GlideOAuthClientResponse: Use these methods for handling client responses.
- GlideOAuthToken: Use the GlideOAuthToken methods for retrieving the access token and information about the access token.

You can also customize the OAuthUtil script include to intercept the request parameters and also parse the responses from external OAuth providers.

OAuth supports application scoping by providing access to all OAuth methods in the available OAuth classes using the `sn_auth` prefix. See [Scoped OAuth APIs](#) on page 1968.

**Scoped OAuth APIs**

OAuth supports application scoping by providing access to all OAuth methods in the available OAuth classes using the `sn_auth` prefix.

See the [ServiceNow Developers site](#) for API information.
For example, the following code accesses the requestToken:

```javascript
var oAuthClient = new sn_auth.GlideOAuthClient();
var tokenResponse = oAuthClient.requestTokenByRequest('TestClient', text);
```

You can still access the methods in the global scope by omitting the `sn_auth` prefix. However, methods that access tokens on locally scoped applications must use the prefix.

## Upgrades and conversions

The ServiceNow platform includes tools to help you during and after an upgrade, and also allows you to convert a ServiceNow Express instance to a ServiceNow Service Management enterprise platform.

### See also

**Upgrade to Geneva**

The upgrade process moves your instance to a new ServiceNow release version. Understand the difference between upgrading and patching, release definitions, rollback and backup options, and how to test your sub-production and production instance upgrades.

### Upgrade History module

The Upgrade History module tracks every upgrade that has been made to an instance. Administrators can use the module to resolve upgrade conflicts and optionally to revert customizations to base system versions to take advantage of new features.

An upgrade history record is created for each upgrade that is run. The *Disposition* field logs the action that was taken on each record during an upgrade. To view an upgrade history record, navigate to **System Diagnostics > Upgrade History** and click the upgrade.

### Table 477: Upgrade History record

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Name of the previous .war file (version).</td>
</tr>
<tr>
<td>To</td>
<td>Name of the applied .war file (version).</td>
</tr>
<tr>
<td>Upgrade started</td>
<td>Time stamp when the upgrade process began.</td>
</tr>
<tr>
<td>Upgrade finished</td>
<td>Time stamp when the upgrade process was completed.</td>
</tr>
</tbody>
</table>

### Record Count by Disposition form section

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inserted</td>
<td>Count of records inserted during the upgrade.</td>
</tr>
<tr>
<td>Updated</td>
<td>Count of records updated during the upgrade.</td>
</tr>
<tr>
<td>Deleted</td>
<td>Count of records deleted during the upgrade.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Skipped</td>
<td>Count of upgrade records skipped (upgrade component not applied) during the upgrade.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To prevent your customizations from being overwritten during system upgrades, the upgrade process skips (does not apply the update to) objects that have been customized. One of your responsibilities as the administrator is to resolve each update that was skipped due to a customization. To resolve a skipped update, you review the reason for each skipped record and then either merge the customization or revert the customization to the base system.</td>
</tr>
<tr>
<td>Skipped error</td>
<td>Count of upgrade records skipped because an error occurred.</td>
</tr>
<tr>
<td>Unchanged</td>
<td>Count of records that were not updated because the baseline component has not changed since the last release.</td>
</tr>
<tr>
<td>Unchanged and Customized</td>
<td>The customer has customized the component. The upgrade component was not applied because the baseline component has not changed since the last release.</td>
</tr>
<tr>
<td>Total</td>
<td>Total count of all records that were affected by the upgrade.</td>
</tr>
<tr>
<td>Upgrade Details</td>
<td>Related list that tracks every record that was affected by the upgrade. Open a record to compare the base system and customized versions.</td>
</tr>
</tbody>
</table>
Review Skipped Records form section

The Upgrade Details list displays each record that was skipped during the upgrade process. Use the list to review the reason for each skipped record in the list and then either merge your customization or revert your customization to the base system.

See Resolve a skipped update on page 1972 for instructions.

Upgrade Details related list

To prevent your customizations from being overwritten during system upgrades, the upgrade process skips (does not apply the update to) objects that have been customized. To assist you in resolving skips, the Upgrade Details form is filtered by default to list all updates that were skipped during the upgrade process.

Navigate to System Diagnostics > Upgrade History to view the Upgrade Details related list.

Table 478: Upgrade Details list

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposition</td>
<td>Action performed on this file during the selected upgrade. Described in Resolve a skipped update on page 1972.</td>
</tr>
<tr>
<td>Priority</td>
<td>Relative importance of the conflict that caused the skip based on the following criteria:</td>
</tr>
<tr>
<td></td>
<td>• 1 (highest priority): xml content</td>
</tr>
<tr>
<td></td>
<td>• 2: script or script_plain</td>
</tr>
<tr>
<td></td>
<td>• 3: html content</td>
</tr>
<tr>
<td></td>
<td>• 4: sys_ui_form_section, sys_ui_related_list, or sys_choice_set</td>
</tr>
<tr>
<td></td>
<td>• 5 (lowest priority): other</td>
</tr>
<tr>
<td>Resolution</td>
<td>• Reviewed</td>
</tr>
<tr>
<td></td>
<td>• Retained</td>
</tr>
<tr>
<td></td>
<td>• Reverted</td>
</tr>
<tr>
<td>Comment</td>
<td>During the process of resolving a skipped update, you have the option to add a Comment to any record. For example, the comment might explain the action that you took to future reviewers.</td>
</tr>
<tr>
<td>Target name</td>
<td>Name of the record corresponding to the current file.</td>
</tr>
<tr>
<td>Plugin</td>
<td>Plugin that contains the record.</td>
</tr>
<tr>
<td>Type</td>
<td>Current file type (such as Business Rule or UI Policy).</td>
</tr>
<tr>
<td>Table</td>
<td>Table that contains the record.</td>
</tr>
<tr>
<td>File name</td>
<td>Current upgrade file name.</td>
</tr>
<tr>
<td>File differences</td>
<td>Comparison of the file in the upgrade with the customized version.</td>
</tr>
<tr>
<td>Changed</td>
<td>Indicates whether the file has changed since the last upgrade. To see this field, configure the form.</td>
</tr>
</tbody>
</table>
Resolve a skipped update

To prevent your customizations from being overwritten during system upgrades, the upgrade process skips (does not apply the update to) objects that have been customized. One of your responsibilities as the administrator is to resolve each skipped update after an upgrade.

Role required: admin

You resolve an update by either retaining the customization or (when the software upgrade contains a feature that you would like to implement) by merging or overwriting the customization with the base system update.

**Note:** Objects that are customized and that did not change in the base system since the last upgrade require no action on your part.

When an object is customized, the system adds a corresponding record to the Customer Updates [sys_update_xml] table and then maintains current version information for all customized objects. The upgrade process skips changes to objects that have a current version in the Customer Updates table. When you follow the procedure, you perform one of the following actions:

- Retain (keep) a customization with no changes
- Retain a customization by merging changes from the updated object
- Revert a customized object to the updated version (that is, overwrite the customization)
- Review the skip and perform no action on the object

1. Navigate to System Diagnostics > Upgrade History.
2. Select the desired software version.
3. In the Upgrade Details related list, select the update record to resolve.

**Note:** By default, the list is filtered by Disposition = Skipped and Resolution Status = Not Reviewed. The Priority value is color-coded with red for P1 and green for P5 priority.

4. Review the list of changes. For text fields, you can click in the field to open the Diff/Merge tool. The File differences field displays a side-by-side comparison of the customization and the update version. Deletions are highlighted red, additions green, and changes yellow. Review the differences. Click a text box to view and edit the detailed differences.

5. Perform one of the following actions. You have the option to add a Comment to any record, for example, to explain the action to future reviewers.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retain the customization with no changes</td>
<td>After reviewing the changes, set Resolution Status to Reviewed and Retained.</td>
</tr>
<tr>
<td>Retain the customization by merging changes from the updated object</td>
<td>Review the differences. To merge a field, click the right-arrow button for the field. Click a text box to view and edit the detailed differences. When you have merged all appropriate fields, click Merge.</td>
</tr>
<tr>
<td></td>
<td>• The Disposition changes from Skipped to Merged.</td>
</tr>
<tr>
<td></td>
<td>• The Resolution Status changes to Reviewed and Merged.</td>
</tr>
</tbody>
</table>
Revert a customization

To prevent customizations from being overwritten by system upgrades, the upgrade process automatically skips changes to objects that have been customized. You may want to overwrite your customizations when a software upgrade contains a feature that you would like to implement.

Role required: admin

To identify customized objects, the system adds a corresponding record in the Customer Updates [sys_update_xml] table. The table maintains the current version information for all objects that have been customized. The upgrade process skips changes to objects that have entries in the table. The upgrade process does not skip objects if only excluded fields have changed.

1. Navigate to System Diagnostics > Upgrade History.
2. Select the desired software version.
3. Filter the Upgrade Details related list by Disposition is Skipped.
4. Optional: Add another filter condition for Changed is True to return only the objects that have changed since the last upgrade.
5. Select the update record to implement.
   The File differences field displays a side-by-side comparison of the customization and the default version. Deletions are highlighted in red, additions in green, and modifications in yellow.
6. Click Revert to base system to overwrite your customized object with the system default version.
   • The Disposition changes from Skipped to Reverted.
   • After you revert a customization, you can click Reapply Changes to reapply your customizations (undo the revert).
Upgrade Monitor

The Upgrade Monitor helps you upgrade an individual instance. You can monitor the progress of an upgrade and resolve conflicts between the upgrade and customizations.

Set up

- Upgrade Monitor overview on page 1974

Use

- Monitor an upgrade to an instance on page 1976
- Process the skipped records list on page 1984
- Resolve conflicts for an individual record on page 1986

Troubleshoot and get help

- Ask or answer questions in the Developer Community
- Search the HI knowledge base for known error articles
- Contact ServiceNow Support

Upgrade Monitor overview

The Upgrade Monitor helps you upgrade an individual instance. You can monitor the progress of an upgrade and resolve conflicts between the upgrade and customizations.

Set up for the Upgrade Monitor

In Geneva, the Upgrade Monitor is installed by default.

If you log in with the admin role while an upgrade is underway, the system automatically displays the Upgrade Progress screen. If no upgrade is in progress, you can navigate to System Diagnostics > Upgrade Monitor.

How the Upgrade Monitor fits into the upgrade process

The Upgrade Monitor concerns only part of the larger upgrade process:

1. Clone the production instance to a test instance and a sub-production instance.
2. Apply the upgrade to the sub-production instance.
3. On the upgraded sub-production instance, process the skipped list.
4. Test the sub-production instance to confirm that the instance still works and performs adequately. Compare to benchmark data from pre-upgrade production instance.
5. Apply the upgrade to the test instance. Import the update sets created on the sub-production instance when you processed the skipped list. Repeat the testing to make sure that the process is working.
6. Apply the upgrade to the production instance. Import the update sets created on the sub-production instance when you processed the skipped list. Test to confirm that the instance works and performs adequately.

Within this larger process, the Upgrade Monitor helps you upgrade individual instances:

- during the upgrade, it shows where in the process the system is
- after the upgrade, it reports what the upgrade did and for how long
- as you upgrade the first sub-production instance, it helps you resolve conflicts between customizations and changes that are part of the upgrade
• on sub-production instances, it provides information that can help you estimate how long the upgrade takes on the production instance.

Monitoring an individual instance as it upgrades

While the upgrade is in progress the Upgrade Progress screen on page 1976 shows what the upgrade process has done, what it is doing, and what remains to be done.

When the upgrade completes, the system displays the Upgrade Complete on page 1978. The Upgrade Summary Report provides information about conflicts between customizations versus the upgrade and provides a link to reconcile these conflicts. For information about understanding and resolving these conflicts, see Process the skipped records list on page 1984.

When you upgrade a sub-production instance, the Upgrade Summary Report can help you estimate how long the same upgrade takes on a production instance. For details about the elements on this report and how to use this information, see Upgrade Details form on page 1986.

Resolving conflicts

To prevent losing customizations, the system skips upgrading records you have customized and provides you with a list of these skipped records.

As you upgrade your first sub-production instance, go through the skipped record list (skipped list) and resolve these conflicts. The system records the changes you make during this process in update sets.

You do not need to reconcile the skipped list on any instances you later upgrade. Instead, you can apply the upgrade then import the update sets containing your changes.

For details on reconciling conflicts, see Process the skipped records list on page 1984.

Factors affecting upgrade duration

Various factors affect how long the system takes to perform an upgrade. The Upgrade Monitor can help you understand those factors and estimate how long the upgrade to your production instance takes.

Many factors can affect the duration of the upgrade process:
• The number of records in the database
• The number of customizations in the database
• The number of nodes in the instance
• The size of tables in the instance that require a schema-change in the instance.
• The number of fix scripts required and the size of the tables those fix scripts manipulate.

Upgrading a sub-production instance can help you estimate how long the upgrade takes on production, but differences between the instances can significantly affect the duration:
• When you cloned the production instance to the sub-production instance, you may have clone-excluded some tables. This reduces the size of the database and makes the sub-production instance upgrade faster than the production instance.
• The production instance may have more memory and processing power.
• The production instance may have more nodes than the sub-production instance.

After upgrading the first sub-production instance, examine the Upgrade Complete on page 1978 for data to help estimate the impact of these factors.
Monitor an upgrade to an instance

While the system is upgrading an instance, you can monitor its progress with the Upgrade Monitor. When the upgrade is done, you can view a summary of the results on the Upgrade Summary Report.

1. If necessary, navigate to System Diagnostics > Upgrade Monitor. When an upgrade is in progress, the Upgrade Progress page shows its status.
2. Monitor the progress of the upgrade. When the system finishes the upgrade, it displays the Upgrade Summary Report.

From the Upgrade Summary Report, resolve any conflicts that prevented the system from upgrading records.

*Upgrade Progress screen*

When an upgrade is underway, the Upgrade Progress screen displays progress bars and other information to help monitor the process.
**Table 479: Screen elements**

<table>
<thead>
<tr>
<th>Screen element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade Progress</td>
<td>A progress bar depicting where the upgrade is in the overall process. The length of a section in the bar does not indicate the relative duration of that process.</td>
</tr>
<tr>
<td>Upgrade Progress: Upgrading Platform</td>
<td>The system is applying upgrades to elements that form the foundation of the platform.</td>
</tr>
<tr>
<td>Screen element</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Upgrade Progress: Updating Schema</td>
<td>The system is scanning the plugins to create a list of tables that require upgrading. This prevents the system from upgrading the same table multiple times.</td>
</tr>
<tr>
<td>Upgrade Progress: Loading Plugins</td>
<td>The system is loading both core and optional plugins. Some features require more than one plugin, so the number of plugins listed may not match the number of optional features installed.</td>
</tr>
<tr>
<td>Upgrade Progress: Completing</td>
<td>The system is upgrading components that need to be completed after the previous three stages are done.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> After the system finishes the Completing phase, it displays a separate screen showing Finalizing. The system tracks in update sets the changes made during the Upgrading Platform, Updating Schema, Loading Plugins, and Completing phases. It does not track changes made during Finalizing.</td>
</tr>
<tr>
<td>Details</td>
<td>Shows the current activity, a progress bar for the current activity, and the file currently being updated.</td>
</tr>
<tr>
<td>Nodes</td>
<td>The system represents the nodes to upgrade as colored squares. The color of a square shows the status of that node during this upgrade:</td>
</tr>
<tr>
<td>Started</td>
<td>Shows when the upgrade started and how long it has been running.</td>
</tr>
</tbody>
</table>

**Upgrade Complete**

The Upgrade Complete screen summarizes the actions taken, provides links to tools for resolving conflicts between customizations and the upgrade, and provides information to help estimate time for upgrades to other instances.
## Upgrade Complete

**Upgrade complete from version 16000 to version 16000 in 35 Minutes**

### Key Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Started</td>
<td>2015-12-30 12:58:58</td>
</tr>
<tr>
<td>Finished</td>
<td>2015-12-30 13:34:22</td>
</tr>
<tr>
<td>Duration</td>
<td>35 Minutes</td>
</tr>
</tbody>
</table>

### Summary of Files

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipped</td>
<td>10</td>
</tr>
<tr>
<td>Updated</td>
<td>36791</td>
</tr>
<tr>
<td>Inserted</td>
<td>553</td>
</tr>
<tr>
<td>Deleted</td>
<td>49</td>
</tr>
<tr>
<td>Unchanged</td>
<td>0</td>
</tr>
<tr>
<td>Unchanged and Customized</td>
<td>0</td>
</tr>
<tr>
<td>Skipped Due to Error</td>
<td>47</td>
</tr>
</tbody>
</table>

*Click here to review updates that were skipped*
<table>
<thead>
<tr>
<th>Screen element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade Complete (summary in green)</td>
<td>The status of the upgrade, including versions.</td>
</tr>
<tr>
<td>Nodes</td>
<td>The system represents the nodes to upgrade as colored squares. The color of a square shows the upgrade status of that node:</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Node Status Icons" /></td>
</tr>
<tr>
<td>Key Metrics</td>
<td>Start time, finish time, and duration of the upgrade.</td>
</tr>
<tr>
<td>Summary of Files</td>
<td></td>
</tr>
<tr>
<td>Key Metrics</td>
<td>The number of records the system did not upgrade because of conflicts between customizations and the upgrade.</td>
</tr>
<tr>
<td>Updated (3)</td>
<td>The number of records the system successfully updated.</td>
</tr>
<tr>
<td>Inserted (4)</td>
<td>The number of records the system inserted.</td>
</tr>
<tr>
<td>Deleted (5)</td>
<td>The number of records the system deleted.</td>
</tr>
<tr>
<td>Unchanged</td>
<td>The number of records in the instance unchanged by the upgrade.</td>
</tr>
<tr>
<td>Unchanged and Customized</td>
<td>The number of customized records in the instance unchanged by the upgrade.</td>
</tr>
<tr>
<td>Skipped Due to Error</td>
<td>The number of records the system did not upgrade because of one or more errors.</td>
</tr>
<tr>
<td>Click here to review updates that were skipped</td>
<td>Click to reconcile conflicts that caused the system to skip some updates.</td>
</tr>
</tbody>
</table>
### Schema Changes to Clone Excluded Tables

<table>
<thead>
<tr>
<th>Table Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>usageanalytics_count_cfg</code></td>
</tr>
<tr>
<td><code>ecc_queue</code></td>
</tr>
<tr>
<td><code>ecc_agent</code></td>
</tr>
<tr>
<td><code>sys_email_account</code></td>
</tr>
<tr>
<td><code>usageanalytics_count</code></td>
</tr>
<tr>
<td><code>sys_user_session</code></td>
</tr>
</tbody>
</table>

Click here to review clone excluded tables

---

**Figure 533: Upgrade Summary Report: Schema Changes to Clone-excluded tables**

The Schema Changes to Clone-excluded Tables section shows a list of tables affected by the upgrade that were clone-excluded when you cloned the production instance to this instance. Because clone-excluded tables are empty, upgrading them takes less time than upgrading those same tables on the production instance. To estimate how much longer the production upgrade takes, note the size of the clone-excluded tables on the production instance.
## Top 10 Fix Scripts by Duration

<table>
<thead>
<tr>
<th>Duration</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 Seconds</td>
<td>_z_metadata_conversion_6.xml</td>
<td>Metadata conversion re-parenting</td>
</tr>
<tr>
<td>7 Seconds</td>
<td>activate_amb_plugin.xml</td>
<td>Load the AMB plugin early, to ensure feature upon first upgrade.</td>
</tr>
<tr>
<td>5 Seconds</td>
<td>_za_populate_table_aliases.xml</td>
<td>Populate existing table aliases on upgrade</td>
</tr>
<tr>
<td>4 Seconds</td>
<td>fix_related_list_calculated_name.xml</td>
<td>Fix sys_ui_related_list.calculated_name</td>
</tr>
<tr>
<td>4 Seconds</td>
<td>activate_live_upgrade_plugin.xml</td>
<td>Load the Live Upgrade plugin early, to ensure feature upon first upgrade.</td>
</tr>
<tr>
<td>2 Seconds</td>
<td>_zb_fix_table_aliases.xml</td>
<td>Fix existing table aliases on upgrade</td>
</tr>
<tr>
<td>1 Second</td>
<td>fix_duplicate_tables_categories.xml</td>
<td>Remove duplicate sys_db_object and invalid sys_number entries</td>
</tr>
<tr>
<td>1 Second</td>
<td>fix_sys_dictionary_duplicates.xml</td>
<td>remove duplicate sys_dictionary records with identical name/element values, keep the eldest in each set of duplicates</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>sys_script_fix_5bf2df13372f1020022a22f3c8e41f15f.xml</td>
<td></td>
</tr>
<tr>
<td>0 Seconds</td>
<td>fix_reparent_listener_detail.xml</td>
<td>Re-parenting sys_listener_detail to sys_metadata</td>
</tr>
</tbody>
</table>

[Click here to see all fix script durations]
The Top 10 Fix Scripts by Duration helps you understand which fix scripts required the most time.

### Top 10 Schema Changes by Duration

<table>
<thead>
<tr>
<th>Duration</th>
<th>Table name</th>
<th>Alter Type(s)</th>
<th>Element Names</th>
<th>Row Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Second</td>
<td>cmdb</td>
<td>modify_column</td>
<td>'serial_number' VARCHAR(255)</td>
<td>2749</td>
</tr>
<tr>
<td>1 Second</td>
<td>sysevent_in_email_action</td>
<td>modify_column</td>
<td>'required_roles' VARCHAR(255)</td>
<td>13</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>cmdb_ci</td>
<td>modify_column</td>
<td>'model_number' VARCHAR(255)</td>
<td>2749</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>sys_schema_attribute</td>
<td>modify_column</td>
<td>'remove_roles' VARCHAR(255)</td>
<td>192</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>question_choice</td>
<td>batch_alter</td>
<td>question_choice</td>
<td>137</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>metadata_clone</td>
<td>create_table</td>
<td>metadata_clone</td>
<td>0</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>cmdb</td>
<td>create_indexes</td>
<td>[serial_number][serial_number]</td>
<td>2749</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>sys_schema_attribute</td>
<td>modify_column</td>
<td>'read_roles' VARCHAR(255)</td>
<td>192</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>cmdb_ci_appl</td>
<td>modify_column</td>
<td>'tcp_port' VARCHAR(255)</td>
<td>16</td>
</tr>
<tr>
<td>0 Seconds</td>
<td>cmdb_ci</td>
<td>modify_column</td>
<td>'ip_address' VARCHAR(255)</td>
<td>2749</td>
</tr>
</tbody>
</table>

[Click here to see all schema change durations]

Figure 535: Upgrade Summary Report: Top 10 Schema Changes by Duration
The Top 10 Schema Changes by Duration helps you understand which schema changes required the most time.

Process the skipped records list

If you customized a record affected by this upgrade, you must resolve the differences between the upgraded and customized versions of the record. This process is called processing the skipped record list or just processing the skipped list.

Role required: admin

1. If necessary, navigate to System Diagnostics > Upgrade Monitor.
   If the upgrade is still in progress, the system displays the Upgrade Progress screen. When the upgrade finishes, the system displays the Upgrade Summary Report.

2. After the system displays the Upgrade Summary Report, click the Click here link in the Skipped box.
   The system displays the System Upgrades form.

3. Navigate to Review Skipped Records section and – if necessary – scroll to the Upgrade Details related list (skipped list).

4. Click the row for the first record you want to reconcile.
   The system displays the Upgrade Details form for that record.

5. Evaluate how you want to resolve the conflict for this record and take the appropriate action:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leave on the skipped list for a later decision.</td>
<td>From the Resolution list, choose Not reviewed to defer the decision on how to handle this conflict. The record stays on the skipped list.</td>
</tr>
<tr>
<td>Leave on the skipped list for a later decision, but record that you have reviewed the record.</td>
<td>From the Resolution list, choose Reviewed. You can filter the list to show only Reviewed records and decide how to handle these records.</td>
</tr>
<tr>
<td>Compare the differences and manually resolve conflicts.</td>
<td>Click Resolve Conflicts to navigate to the Resolve Conflicts form on page 1988.</td>
</tr>
<tr>
<td>Leave the customized record as is and do not update it.</td>
<td>From the Resolution list, click Reviewed and Retained.</td>
</tr>
<tr>
<td>Discard the customization and update the record to match the base system for this upgrade</td>
<td>Click Revert to Base System.</td>
</tr>
</tbody>
</table>

Note: The system tracks changes to records in an update set so you can apply these changes to another instance later. However, the system does not migrate the upgrade details records from one instance to the next. These records apply to a specific upgrade of a specific instance. If you want to preserve the Comments, Resolutions, or other information from the skipped list, export it from this instance.

6. In the Comment field, write the reasons for making your decision and other information you want to document.

7. Click Update.

System Upgrades form

When an upgrade is complete, the System Upgrades form displays key statistics about the upgrade and a related list of skipped records (the skipped list).
Figure 536: System Upgrades form

Table 481: Fields

<table>
<thead>
<tr>
<th>Screen element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>The version of the instance before upgrading.</td>
</tr>
<tr>
<td>To</td>
<td>The version of the instance after upgrading.</td>
</tr>
<tr>
<td>Upgrade started</td>
<td>The time upgrade started.</td>
</tr>
<tr>
<td>Upgrade finished</td>
<td>The time the upgrade finished.</td>
</tr>
<tr>
<td>Skipped</td>
<td>The number of records the system did not upgrade because of conflicts between customizations and the upgrade.</td>
</tr>
<tr>
<td>Unchanged</td>
<td>The number of records in the instance unchanged by the upgrade.</td>
</tr>
<tr>
<td>Unchanged and customized</td>
<td>The number of customized records in the instance unchanged by the upgrade.</td>
</tr>
</tbody>
</table>
## Resolve conflicts for an individual record

Reconcile differences between your customized record and the changes associated with the upgrade.

**Role required:** admin

1. From the [Upgrade Details form](#) for the record you are reconciling, click **Resolve Conflicts**.
   
   The system displays the [Resolve Conflicts form](#), which highlights differences between the two versions of the record. The form displays information about the base system record on the left and the customized record on the right.

2. Compare the base system with the customized record for each field on this form. For non-script fields, edit the customized record on the right-hand side to include what you want from the base system and the customization.

3. If this record contains a script, check it for conflicts and resolve.
   
   a) Click inside the Script field.
      
      The system displays the Resolve Conflicts - Script form highlighting areas where the two versions of the script differ.

   b) Edit the right-hand side so that the script contains whichever content you want. To move a block of code from the left to right side, click the small arrows corresponding to that block in the middle column.

   c) Click **OK**.
      
      The system returns to the Resolve Conflicts form.

4. To save your changes to the record, click **Save Merge**.
   
   The system sets the Resolution for this record to **Reviewed and Merged**.

---

### Upgrade Details form

From the Upgrade Details form, you can review an individual record affected by the upgrade and reconcile conflicts between the upgrade and customizations.
Geneva    ServiceNow    ServiceNow Platform

**Figure 537: Upgrade Details form**

<table>
<thead>
<tr>
<th>Screen element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name</td>
<td>The record the system has flagged as needing to be reconciled.</td>
</tr>
<tr>
<td>Priority</td>
<td>The priority the system has assigned to resolving this conflict. Values range from one to five, with one representing the highest priority.</td>
</tr>
<tr>
<td>Comment</td>
<td>Comments to document your decisions about reconciling this record.</td>
</tr>
<tr>
<td>Screen element</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Resolution     | How you elected to resolve this conflict:  
  • Not Reviewed  
  • Reviewed - reviewed but no action yet taken  
  • Reviewed and Merged - made changes to the record to reconcile the customized and upgraded versions  
  • Reviewed and Retained - left customizations in place without update from upgrade  
  • Reviewed and Reverted - customizations discarded, record updated according to upgrade  
  For more information, see Process the skipped records list on page 1984. |
| Disposition     | The action the system took on this record. |
| Type            | The record type, for example Script include. |
| Target name     | Name of the skipped record, if applicable. |
| Update set      | Unused. |
| Plugin          | The plugin containing this record. |
| Table           | The table containing this record. |

Resolve Conflicts form
The Resolve Conflicts form you compare to the base system version with the customized version of a record and reconcile the differences.
Figure 538: Resolve Conflicts form

### Fields
The fields this form displays depend on the type of record you are reconciling.
The left column shows the records fields in the base system, including the proposed changes that are part of the upgrade. The right column shows the fields for your customized record.

### Express conversion to ServiceNow Enterprise
When a ServiceNow Express instance is converted to the ServiceNow Service Management enterprise platform, the instance is changed significantly.
The following topics help you understand what to expect during and after you convert from the ServiceNow Express platform to the ServiceNow Service Management enterprise platform. The topics provide advice and recommendations for advanced features and applications for you to consider.

It is recommended that you read the topics in the following order.

**Express conversion to Enterprise overview**

As you plan your upgrade from the ServiceNow Express platform to the ServiceNow Enterprise platform, become aware of the many features and resources that are available to you, both inside and outside of your instance.

Your upgraded instance will have significant new functionality that is part of the standard service management suite, including the following.

- **Graphical workflow**: Simplify complex service catalog requests with the highly configurable drag-and-drop workflow editor.
- **Scripting**: Extend your instance beyond standard configurations.
- **Content management**: create a custom interface for the ServiceNow platform and ServiceNow applications.

In addition to the standard service management suite, there are additional applications and suites that your organization can purchase, for example, financial and human resources applications, and the operation management suite. The ServiceNow Express platform did not offer these solutions, and it is recommended that you explore the resources available to learn about extending your ServiceNow Enterprise instance.

Where you may have been using the Express support site forums for help, a larger Enterprise community site is now available.

There are many resources available on the community site and on YouTube, but the ServiceNow Demo Center is the best resource to quickly learn about Enterprise platform functionality. The Demo Center offers on-demand videos and live presentations where you can ask questions via chat.

*Click here to access the Demo Center*. Enter email address you registered with ServiceNow as your username, and reset your password if you are attempting to access the Demo Center for the first time.

Recommended Demo Center videos for an organization that upgrades from the Express platform to the Enterprise platform include the following:

- **Skinning the self service portals via Content Management**
- **Asset management life cycle**
- **Creating automation via Workflows**

**Express conversion to Enterprise process**

The conversion process includes an opportunity for you to review and test the ServiceNow Enterprise platform with a copy of your data before your production instance is converted.

**Conversion process**

The conversion process follows this timeline

1. A sub-production instance is created from a cloned copy of your ServiceNow Express production instance. This sub-production instance is for the full Enterprise Service Management platform.
2. You receive a letter indicating that you are entering a 15-day testing phase. The purpose is to verify that your ITIL and self-service portal users continue to experience the behavior they are familiar with.
3. After you verify with the sub-production instance that the conversion completed and is acceptable, your designated Hi Administrator updates the change record advising ServiceNow support of your acceptance.

4. ServiceNow support schedules a full conversion of your Express production instance to ServiceNow Service Management. Your sub-production instance is paired to it, and is available for you to configure new features, validate them, and roll them out to production.

ESS and ITIL experience

After an Express instance is converted to the Service Management platform, users with ESS and ITIL roles experience the look, feel, and behavior they are familiar with. They can access the same sets of applications and modules, lists and forms, and they experience the same underlying behavior. All functions that were accessible to ESS and ITIL users in the Express instance look and behave the same way for all users after the conversion.

For example:

- The incident form looks and works exactly the same way as it did in the Express instance. The same is true for the Problem, Change Request, and Configuration Item forms.
- Approvals are requested the same way as in the Express instance, and the approval process is the same.
- Survey responses are generated as in the Express instance.

Administrator experience

The Service Management platform is significantly different for administrators, who have access to all of its features. It is recommended that administrators attend the ServiceNow System Administrator training.

Express conversion to Enterprise considerations

Consider four core areas of functionality when planning your upgrade from the ServiceNow Express platform to the ServiceNow Enterprise platform. In these areas, Enterprise platform capabilities are more configurable, powerful, and flexible than in the Express platform.

Approval rule to workflow

As part of your migration from the Express platform to the Enterprise platform, reconsider your approval rules.

- In the Express platform, approval rules are applied to a record based on conditions when the record is saved. Approval rules run only when requested, and always occur first for catalog requests.
- In the Enterprise platform, approvals are an activity that is generally defined within the graphical workflow. This provides enhanced flexibility as you can define approval loops, re-approvals, and approvals that are nested at the beginning or end of a workflow. Reminder tasks can be defined to remind approvers that they need to take action.
Service level agreement (SLA) engine

The Express SLA engine is also record-based. In the Enterprise platform, the SLA engine has additional features, such as the ability to schedule jobs to auto-calculate time passed or time remaining metrics. In addition, workflow can be used to configure additional SLA complexity.

Security

The Express platform uses simple security for create, read, update, delete (CRUD) access through the dictionary. The Enterprise platform uses contextual security for more robust access control, referred to as access control lists (ACLs) or access control rules. During the conversion, your CRUD settings are converted to ACLs. The conversion does not change any of your security access.

The Enterprise platform offers additional benefits in the area of security. Security-based plugins are available and provide column level encryption, edge encryption, or encryption for data at rest.

There are also a greater number of roles available in the Enterprise platform that provide more granular options for granting access to the appropriate users to take action. Going forward, you will manage security settings using ACLs. For more information, see Access control rules on page 2498 and Platform security on page 2435.

Development Instance

On the Enterprise platform, you can utilize a development instance in addition to your production instance. The development instance gives you a place to test and validate new Enterprise configurations. As you make changes to the development instance, you can use update sets to track and push the configurations to your production instance.

Scripting and other configuration options

As an administrator of a ServiceNow Enterprise instance, you have more configuration options available than in an Express instance. For example, there are additional field types on forms and an extensive list of dictionary options for fields or tables.

Administrators have access to core ServiceNow platform features such as scripted business rules, client scripts, UI actions, UI pages, UI macros, UI scripts, script includes, script actions, and so on. Administrators have access to all tables and dictionary entries, and can use scripting to accomplish any change in the instance. Examples of custom configurations include the following.

- You can use UI actions on page 734 to define custom buttons.
- You can use Client scripts on page 3900 to define client-side scripts to run calculations or change fields dynamically before the form is saved.

Revisit your Express process as regards workflow and configuration in the context of how the Enterprise platform works. Determine ways to optimize your process based on Enterprise platform flexibility.

Enterprise plugins

The ServiceNow Enterprise platform offers enhanced functionality via plugins, and new plugins become available with each new release. When you upgrade, most plugins are not enabled on your converted Enterprise instance. This is slightly different from the plugins activated by default on a new Enterprise instance.
After the conversion, the administrator can navigate to **System Definition > Plugins** and review the list of plugins that are available and active. The administrator can also choose which capabilities to **activate after the conversion**. The **List of Geneva plugins** on page 1234 lists the available plugins.

**List of plugins to consider activating after a conversion**

After the conversion is completed, review the plugins that are available and choose the capabilities to activate after. The plugins listed below are not activated on an Express to Enterprise platform conversion. Learn about the functionality these plugins provide and determine which ones to activate.

For a description of plugins not linked below, review the description in the **List of Geneva plugins** on page 1234.

1. Assessments (com.snc.assessment_core)
2. Asset Management (com.snc.asset_management)
3. Risk assessment: Best Practice - Change Risk Calculator (com.snc.bestpractice.change_risk)
4. Incident resolution and recovery: Best Practice - Incident Resolution Workflow (com.snc.bestpractice.incident)
5. Best Practice - ITIL KPI Reports (com.snc.bestpractice.itil_kpi)
6. Survey wizards: Best Practice - Task Survey Management (com.snc.bestpractice.task_survey)
7. Catalog Designer Common (com.glide.ui.ng.cc)
9. Change Management - Core (com.snc.change_management)
11. Contract Management (com.snc.contract_management)
12. Problem Management – Create knowledge: Create knowledge from problem (com.snc.problem_kb)
13. Data lookup and record matching support for Service Catalog (com.glide.data_lookup.catalog)
14. Depreciation (com.snc.depreciation) installed with Asset Management
15. Expense Line (com.snc.expense_line)
16. Create fixed assets (com.snc.fixed_asset)
17. High Security Settings (com.glide.high_security)
18. Planned Maintenance: Maintenance Schedules (com.snc.maintenance_schedules)
19. Model management (com.snc.model)
20. My Assets (com.snc.asset_myassets)
21. Organization Management (com.snc.organization_management)
22. Performance Analytics concepts (com.snc.pa)
23. Problem Tasks (com.snc.problem_task)
25. Role Delegation (com.snc.role_delegation)
27. Service Creator (com.glide.service_creator)
28. Service Level Management (com.snc.sla)
29. Software Asset Management Extensions (com.snc.sam)
30. Survey designer (com.glide.survey_designer)
31. System user guide (com.glide.user_guide)

Additional ServiceNow features and applications

There are a large number of additional ServiceNow applications available for purchase after your conversion from the Express platform is complete.

The following list include some examples of these applications. Contact your sales representative for more information.

• Service Mapping
• Password Reset
• Automated Software or Configuration Deployment
• Automated External Script Execution
• Event Management
• Security and Vulnerability Management
• Cloud management
• Advanced reporting through performance analytics
• HR Service Management
• Project and Portfolio Management

Important features in the base system

Review the features listed below for an understanding of more ways to extend platform functionality.

• Update sets
• Service level agreements
• Workflow
• Project Management
• SDLC scrum process
• Import sets
• Transform maps on page 1528
• LDAP
• Content Management System on page 2040

Additional resources

A number of additional resources are available to learn more about the Enterprise platform. As mentioned, the Enterprise community is similar to Express forums, and it provides help and support to answer all of your questions. The community includes regional user conferences that are referred to as SNUGS (ServiceNow User Groups). Joining a SNUG is a great way to network and learn from other businesses that use the ServiceNow Enterprise platform.

Since the Enterprise platform comes with functionality to download and install update sets of external configurations, it is popular for users to share applications and configurations they have built via the Share portal.

Since the Enterprise platform has a large number of additional integration points, more pre-built content is available in the ServiceNow App Store. Certified partners are members of our large technology partner
community, and through the App Store they offer popular applications and integrations to help you integrate with your 3rd party systems. The App Store is an excellent resource for acquiring additional functionality outside of ServiceNow applications and suites.

Platform performance

The perceived performance of your ServiceNow instance is made up of certain components. Troubleshooting performance by monitoring and diagnosing the following components helps to ensure system optimization.

- Application server response: Time for the application server to process a request and render the resultant page.
- Network latency and throughput: Time for the network to pass your request to the server and to pass the response back.
- Browser rendering and parsing: Time for your browser to render the HTML and parse/execute Javascript.
- Instance cache: The amount of system resources available for processing.

Instance cache effects on performance

There is a performance degradation whenever you purge and rebuild the instance cache. During core business hours, avoid or minimize any of the following actions that cause a purge and rebuild of the instance cache.

- Adding or updating system properties
- Adding or updating dictionary entries
- Committing update sets
- Adding or updating translations

Transaction log response times

The instance automatically logs the vital statistics of every transaction it processes, and that information is available to you as an administrator.

To look at the log, navigate to System Logs > Transactions.

To see the average response time of all listed transactions, right-click the column Response time. Select Configure > List Calculations, and then select the Average value check box.

It is a good practice to limit the list to those transactions that took place during the time period of interest. The default filter returns transactions from today.
Figure 539: Transaction log

For each completed transaction, the following information is available (times are in milliseconds):

- Date/time, User ID, IP address, and URL of the transaction.
- Total response time, which does not include the browser time because the server does not have that information.
- **Network time**: network transmission time, both from and to the user.
- **SQL time**: time spent executing SQL commands.
- **SQL count**: number of SQL commands executed.
- **Business rule time**: time spent processing business rules.
- **Business rule count**: number of business rules executed.
- **Output length**: how many bytes the transaction returned, after any compression.

Response time on forms

A response time indicator (clock) appears at the bottom right of many forms and lists.

This indicator provides the processing time for a completed transaction, including the total time and the time for each step. Click the icon to show and hide the response time details. Point to the icon to see a tooltip with the details. The following example shows the response time for retrieving a filtered list in a demo instance.

![Response time](image)

**Response time(ms): 1019, network: 3, server: 526, browser: 490**

Figure 540: Response time

In this example, the transaction took the following amount of processing time:

- 1019 milliseconds total time
- 3 milliseconds moving data across the network
• 526 milliseconds on the server
• 490 milliseconds in the browser, rendering the HTML and parsing and executing JavaScript

Response does not appear for simple operations, such as paging, changing a list sort order, or for the first transaction in a session.

To view a detailed breakdown of the browser processing time on forms, click browser.

![Figure 541: Browser response time details](image)

Administrators can disable the response time by setting the glide.ui.response_time property to false.

When you review response times, look for the following issues.

1. A period where all transactions took an unusually long time. For example, transactions that normally took 1 second took 15 seconds between 11:00 AM and 11:20 AM. This issue can indicate that an unusual load, such as a large report, ran on that app server during that time.

2. A specific transaction which repeatedly took an unusually long time. For example, the list of closed incidents sorted by short description took 30 seconds each time it was displayed. This issue can indicate that a particular transaction put an unusual database load on the system, such as sorting 500,000 records on an unindexed field.

Try the following actions to improve performance.

1. Look for one or more transactions that span the entire window. For example, you observe that the response was slow for six minutes and one transaction ran the entire time. You can try adding additional indexing to the database to make the transaction faster. Certain types of queries are always going to run more slowly than others, regardless of indexing.

2. Ensure that a cache flush is not being run during business hours. Cache flushes are intended to prevent older data from interfering with changes and updates, and are performed automatically when using update sets. Scheduled cache flushes, using cache.do, can affect overall performance, and degrade system response times. Do not run cache flushes during business hours, and do not trigger cache flushes automatically.

3. If you cannot find any specific issues when experiencing slow response time, contact support to see if there are global issues with the application server hardware.

**Client transaction timings**

The Client Transaction Timings plugin provides extra information about the amount of time spent on both the client and server side, and by the browser and network. This feature not only helps find long-running processes, but also provides information about where in the process the performance issue is caused.
Network response times

Troubleshooting a poor network response time can be difficult, but there are certain quick tests you can perform.

One clear indicator of a network issue is when you find that users in one location have good performance, and users in another location have poor performance. That tells you that the server and application are fine. Assuming that browser settings are identical, the only meaningful difference is the network.

Ping times

The coarsest measure of network response time is a ping. A ping measures the total time for a packet to make it from the source machine to the target and back again.

To perform a ping in Windows, bring up a command window (DOS prompt) and type:

```
ping -t <yourinstancename>.service-now.com
```

Look for a time under 100 ms if you are in the U.S., or 150 ms if you are in Europe or Asia. In practice, anything less than 250 ms is not of concern as it is not generally a major component in your perceived response time.

Traceroute

If you are seeing slow ping times, you can run a traceroute. Some networks refuse to forward ICMP, and your traceroute request may not work. If it does work, it is a great tool for identifying network bottlenecks.

To run a traceroute on Windows, bring up a command window and run the following command:

```
tracert <yourinstancename>.service-now.com
```
Sample output:

```
C:\dev\mysql5\bin>tracert mycompany.service-now.com
Tracing route to mycompany.service-now.com [70.87.98.130]
over a maximum of 30 hops:
1     1 ms     1 ms     1 ms  12.192.116.193
2     4 ms     4 ms     4 ms  12.116.227.37
3    32 ms    32 ms    32 ms  gbr1-p90.sd2ca.ip.att.net [12.123.145.178]
4    33 ms    33 ms    33 ms  tbr1-p013503.phmaz.ip.att.net [12.122.2.142]
5    34 ms    34 ms    34 ms  tbr2-cl1521.phmaz.ip.att.net [12.122.10.194]
6    32 ms    32 ms    32 ms  tbr2-cl1592.dlstx.ip.att.net [12.122.10.81]
7    31 ms    50 ms    31 ms  gar1-p370.dllrtx.ip.att.net [12.123.16.173]
8    31 ms    31 ms    31 ms  12.119.136.14
9    31 ms    31 ms    31 ms  te9-1.dsr02.dllstx3.theplanet.com
[70.87.253.22]
10    37 ms    37 ms    37 ms  vl41.dsr01.dllstx4.theplanet.com
[70.85.127.83]
11    31 ms    37 ms    31 ms  gil-0-1.car16.dllstx4.theplanet.com
[67.18.116.67]
12    32 ms    32 ms    32 ms  70.87.98.130
Trace complete.
```

Each line in the traceroute represents a network step between the source machine and the destination machine. In the sample traceroute, there were a total of 12 steps required to get the network traffic from the laptop to <yourinstancename>.service-now.com.

- The left column is the step number.
- The next three columns are latency estimates, performed three times to give an average.
- The last column is the machine you are hopping to.

For example, from rows #1 and #2 above, you can tell:

```
1     1 ms     1 ms     1 ms  12.192.116.193
2     4 ms     4 ms     4 ms  12.116.227.37
```

At the end of row 1, it was at 12.192.116.193. It then took 4 ms (on average) to get to 12.116.227.37.

Generally, with a traceroute, you are looking for individual steps that take a long time, like 500 ms for a particular hop. You are also looking for steps that show an asterisk (*) instead of a step time, for example:

```
1     100 ms   *        500ms  12.192.116.193
```

The asterisk indicates that a particular packet failed to arrive, which can indicate network problems on that particular hop. You also see an asterisk if that particular router is set to not forward ICMP. This outcome is potentially a false alarm if all three latency times for a step are asterisks.

**Browser settings and performance**

Depending on the browser you use, browser settings can affect the performance of your instance.

**Compression**

Web pages in your instance can be large, for example, over 500 kb for a long list of incidents with many columns. To speed performance, most browsers can accept compressed data from an application server so that the full 500 kb does not have to be sent over the wire. Instead, the browser indicates that it can accept compressed data if the server can send it. The app server then compresses the response, which transforms the 500 kb data to about 20 kb.
Compression is enabled by default on all ServiceNow application servers. The application server always sends compressed data if your browser accepts it. There are browser settings that control whether your browser accepts compressed responses.

In Microsoft Internet Explorer (IE), navigate to Tools > Internet Options. Ensure that the following two check boxes are selected in the Advanced tab.

- Use HTTP 1.1
- Use HTTP 1.1 through a proxy server

![Internet Options](image)

**Figure 542: Internet options**

Frequently, a proxy or edge device in the environment disables gzip compression. Enabling gzip compression also speeds up the interactions.

**Cache items from https locations**

If you enforce an IE policy to never cache items from an https location, every interaction must refetch a large amount of JavaScript and images from the server, which affects performance. The IE option is **Do not save encrypted pages to disk**, and the default is off.

**Database performance tuning tools**

To aid in performance evaluation, the Stats Tools diagnostics plugin records statistics for system activities that affect performance. System activities include execution of queries, scripts, and transactions.

**Note:** The Stats Tools plugin is activated by default. It requires the admin role to activate or upgrade, and it requires the com.glide.monitor.round_robin_database.

Stats Tools adds modules under **System Diagnostics > Stats**, including **Slow Queries**, **Slow Scripts**, and **Slow Transactions**. Each module accesses a table of activity patterns [sys_query_pattern], [sys_script_pattern], [sys_transaction_pattern]. Each pattern table represents a collection of unique activities. Each collection is an aggregation of executions of that unique activity over all time. Each record provides basic timing analysis with example identifiable details of the activity.

Activity patterns are immediately recorded to a cache and are later persisted to their pattern table. If you flush server caches, then recorded activities that have not been persisted are cleared.

Following are examples of pattern records.

- Each time a query is executed that meets the recording and persistence threshold it is aggregated and stored as a query pattern record.
- Each time a particular business rule is executed it aggregates to a script pattern record.
• Each time a particular background job runs it aggregates into a unique transaction pattern record.
• Each click of the **New** button on the Incidents list counts as a list type transaction pattern with specific form action.

**Metrics**

Metrics include total and average times of interest per unique execution pattern over the total execution count. Metrics are aggregated with each new instance of the unique activity and persisted to the pattern record.

**Metadata**

Example data from specific executions are included on each pattern to identify execution details.

**Characterizations of each activity type**

| Transactions | Any transaction type includes server-side and related client-side transactions. Metrics include **Total server load time**, which aggregates the total server-side time excluding semaphore and session wait times. It also aggregates relevant server transaction times that are found on the syslog_transaction table. Transaction types:
|              | • An HTTP Request transaction is identified by a URL, transaction type, processor, form/list action, URL query (filters), and related table name.
|              | • Any other transaction is identified by its transaction URL/page/name, transaction type, and processor or thread name. |
| Scripts      | Any script activity type includes scripts evaluated by GlideScopedEvaluator. Script Types:
|              | • A Jelly Script is identified by the sys_jelly_file table, jelly file path, line number, and script that was executed.
|              | • Any other script is identified by the table and sys_id. |
| Queries      | Any query activity includes prepared statements executed by GlideDBI Query Types:
|              | • All queries are identified by MongoDB query or insert, update, or select statements. They |
Use a slow query log

Administrators can use slow query logs to gain insight into how queries are affecting platform performance. The Slow Queries log aggregates data for similar queries. Use slow query data to evaluate the need for new indexes, changes to existing indexes, or changes to frequent queries.

Role required: admin

Queries are similar when they select from the same table and query the same field in the where clause, but search for different values in the field. For example, these queries are aggregated as similar queries.

```
SELECT * FROM sys_user WHERE name="ITIL User"
SELECT * FROM sys_user WHERE name="System Administrator"
```

Aggregating similar query data allows administrators to monitor the performance impact of slow queries and of queries that occur frequently. The log reports data for similar queries where the total execution time exceeds 5 seconds.

1. Navigate to System Diagnostics > Stats > Slow Queries

2. Open a query record for more details.
Table 483: Query record fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>An SQL statement for an individual query, before being canonicalized to aggregate it with similar queries.</td>
</tr>
<tr>
<td>Example stack trace</td>
<td>A stack trace for an individual query. Lines referencing script code (includes both custom and base system code) appear in the following format:</td>
</tr>
<tr>
<td></td>
<td><code>table name.sys_id:line number</code></td>
</tr>
<tr>
<td></td>
<td>For example, this stack trace indicates a function call from line 119 of a script include.</td>
</tr>
<tr>
<td></td>
<td><code>sys_script_include.105f70abc0a8010300d4d79ed1b93eb0:119</code></td>
</tr>
<tr>
<td></td>
<td>For more information, see stack traces.</td>
</tr>
<tr>
<td>Example URL</td>
<td>The URL for an individual query, depending on how the query was called.</td>
</tr>
<tr>
<td></td>
<td>• User transactions: lists the transaction parameters. For example, if a user navigates to the incident list, the URL is <code>/incident_list.do</code></td>
</tr>
<tr>
<td></td>
<td><code>sysparm_query=active=true</code></td>
</tr>
<tr>
<td></td>
<td>• Scheduled jobs: lists the name of the scheduled job.</td>
</tr>
<tr>
<td></td>
<td>• Any other method: lists an empty field.</td>
</tr>
<tr>
<td>First sighting</td>
<td>The first occurrence of a similar query.</td>
</tr>
<tr>
<td>Execution Count</td>
<td>The number of similar query occurrences that are aggregated.</td>
</tr>
<tr>
<td>Total execution time</td>
<td>The sum of execution time for these similar queries.</td>
</tr>
<tr>
<td>Average execution time (ms)</td>
<td>The average duration to execute one of these similar queries.</td>
</tr>
</tbody>
</table>

3. To see what the database is doing to retrieve the data, click Explain Plan. The query plan is reported in the MySQL Explain Plan related list. Use the query plan to evaluate the need for new indexes or changes to existing indexes. For more information, see MySQL explain plan documentation.

This feature works on other supported databases, including MongoDB and Oracle.

Transaction cancellation

A user can cancel a transaction that takes longer than expected to load.
Causes of slow-loading transactions include the following scenarios.

- Sorting a large number of records by an unindexed string field.
- Grouping a large table on a field that has predominantly distinct values.
- Exporting all rows from a large table.
- Testing a poorly scripted business rule that has an infinite JavaScript loop.

**Note:** You cannot cancel an import with these controls.

### Transaction cancel timer

During a long-running transaction, a timer and a red cancellation button appear in the banner frame.

![Figure 543: Cancel request](image)  
To cancel the transaction, click the cancellation button. The timer indicates that the transaction is being canceled. When it is finished, you see a message that the transaction was canceled.

If the transaction completes successfully, the timer indicates how long it took for the transaction to complete.

![Figure 544: Transaction completed](image)

### Canceled transaction logs

Canceled transactions appear in the transaction log with **CANCELLED** appended to the URL. Transactions canceled by a user are logged differently from transactions canceled by [Transaction quotas](#) on page 2032.

### Cancel transaction properties

Administrators can configure the behavior of the transaction cancel capability using the following properties.
Geneva  ServiceNow  ServiceNow Platform

<table>
<thead>
<tr>
<th>Property</th>
<th>Definition</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.glide.request_manager.active</td>
<td>Let users cancel long running transactions (enabled by default).</td>
<td>Open the sys_properties table.</td>
</tr>
<tr>
<td>glide.ui.transaction.long_response</td>
<td>Delay in seconds before the cancel transaction button appears for a long-running transaction.</td>
<td>Navigate to System Properties &gt; UI Properties</td>
</tr>
</tbody>
</table>

**View and kill active transactions**

Administrators can view active transactions and kill long-running transactions if necessary from the Active Transactions module.

Role required: admin

This action is intended to stop background processes, such as scheduled jobs, and not user-initiated transactions.

1. If **high security** is enabled, elevate privileges to security_admin.
2. Navigate to **User Administration > Active Transactions**.

3. Select the transaction you want to stop.
4. Perform one of the following actions.
   - Right-click the record and select **Kill**.
   - Select the check box beside the record and select **Kill** from the **Actions on selected rows** choice list at the bottom of the list.

**Import set performance**

An algorithm is used to transform import sets from their staging table into their final destination.

Importing data via an import set requires the following two phases.

1. The data is loaded from a data source into a staging table.
2. The data is transformed from the staging table into a target table.

The transform algorithm operates in blocks of 100 records at a time. This approach allows the application server to pre-fetch a variety of information relevant to each block of records. This action reduces the number of unique interactions with the database and improves throughput.
Performance is improved for all large transformations are using this algorithm. It also improves transformations with many reference or choice type columns. Transformations that with complex or unkeyed coalesce conditions experience a proportionately smaller benefit from the algorithm.

**Thread performance monitoring**

Performance of individual threads is tracked by the instance. Administrators can view thread performance information.

Thread performance can be monitored in two places as described in the following sections.

**Transaction log**

The amount of time a thread waits for an available semaphore or session synchronization is included in the **Response Time** field in the transaction log. To view the transaction log, navigate to **System Logs > Transactions**.

**Performance graph**

Every second, the system looks at all active threads (both UI and background) and places them into one of the following categories.

- **database**: waiting for information from the database.
- **network**: writing data out to the network or waiting for an outbound network buffer to flush.
- **business_rule**: the system is running a business rule (synchronous or asynchronous) and is not currently executing a query (which would be database).
- **concurrency**: cannot run because they are waiting on a semaphore or session synchronization.
- **cpu**: the thread is active, but is not executing any of the steps. This condition typically means non-business rule compute time, although a few other internal wait states are categorized as CPU in this case. Therefore a 1:1 correlation between threads in a CPU count and hardware CPU utilization is not expected.

Every minute, the system averages these transactions and records them in the database. These averages are displayed in the **System Overview** chart.

![System Overview chart of active threads](image)

**Figure 545: System Overview chart of active threads**

The **System Overview** chart can be added to any homepage. For more information, see **Add existing gauges to a homepage** on page 478.
Performance metrics

You can view a wide range of performance metrics for your instance and for the machine on which your instance is running. The information is displayed in a graphical format.

Add performance metric graphs and their controls to your homepage to monitor the performance of your instances. Some of these graphs are intended for use by ServiceNow Technical Support to troubleshoot performance issues or help you tune your system for maximum efficiency. Each graph enables you to filter the data by using different measurements, such as maximum and minimum values, means, and medians. The available graphs reflect performance in eight functional areas.

The administrator or any user who can modify a homepage can add performance metric graphs to a homepage. For more information, see Add existing gauges to a homepage on page 478 and Homepage customization on page 489.

Replication metrics

Replication is the process whereby an entire instance is replicated on a second machine for failover protection. The Replication Throughput graph measures the difference in the data (delta) between the production instance and the replicated instance as user activity changes the database. ServiceNow Technical Support uses this information to monitor the progress of replicating a customer instance.

Database graphs

Database graphs display metrics for various database operations, for example, insertions and deletions. They also display a current count of database connections for the selected ServiceNow instance.

To view the details of a database, select an instance from the CI list, and then select a database operation to view. Available database operations are listed above the graph. All the traffic listed here is specific to your cluster node/instance. If you are looking at the database graph for your discovery node, it does not include traffic for your UI node or vice versa. If you, like most customers, are running on a single node, the graph shows everything.
Figure 546: Performance metrics database graphs

- **Database Throughput**: Displays the count (per minute) for each type of database operation (inserts, deletes, and so on) over time.
- **Database Response**: Displays the response time (in milliseconds) for each type of database operation.
- **DB Connection Use**: Shows how many connections this instance has open to the database. Select to display as a maximum, a minimum, or a mean.
Check boxes above each graph allow you to control the display of available data.

![Deletes](✓) [Inserts](✓) [Selects](✓) [Updates](✓)

**Figure 547: Available metrics**

Clear a check box to remove the input from the graph.

**Discovery performance metrics**

Discovery metrics measure the performance of the probes and sensors in your instance as they collect information about CIs in the network.

**Note:** Functionality described here requires the Discovery plugin.

The available measurements to apply are **count**, **maximum**, **mean**, **median**, and **minimum**. All times in these graphs are measured in milliseconds.
Figure 548: Discovery performance metrics graphs
• **Discovery Probe Run Time**: Elapsed time, in milliseconds, that probes take to run. Use this chart to check the performance of your MID Server. Slow run times can indicate a resource problem on the MID Server machine.

• **Discovery Sensor Queue Time**: Indicates how long a sensor task sits in the scheduler queue on the instance before it runs.

• **Discovery Sensor Run Time**: Measures how long a sensor task takes to run after it is started.

### Disk partition statistics

Disk partitions let you view the input and output statistics for partitions on the server that hosts your ServiceNow instance.

The available measurements are the read and write requests per minute and the number of bytes read and written per minute.

Set: **Disk Partitions**

![](image)

**Figure 549: Disk partition performance metrics**

You can display the statistics for each partition or the cumulative totals for all partitions. To view cumulative I/O for the entire disk, select the first item in the CI choice list. Select one of the individual partitions from the list to view the statistics for that partition alone.
Figure 550: List of disk partitions

Four of the disk partitions that appear in the list are generated as business services and are used to populate the performance graphs with data. If they are deleted, the instance automatically regenerates them. They appear in the list of business services.

<table>
<thead>
<tr>
<th>Business Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Blackberry</td>
</tr>
<tr>
<td>Bond Trading</td>
</tr>
<tr>
<td>Bond Trading - DR</td>
</tr>
<tr>
<td>bosdemo01.service-now.com:demosw_16245</td>
</tr>
<tr>
<td>Client Services</td>
</tr>
<tr>
<td>E-Commerce</td>
</tr>
<tr>
<td>Electronic Messaging</td>
</tr>
</tbody>
</table>

Figure 551: Business services disk partitions

Linux statistics

The Linux Stats graph displays performance data for the server on which your ServiceNow instance is running.

These graphs are used as a tool to locate and resolve issues with your instance.

- **CPU Usage**: The measurement of this graph indicates the percentage of the available CPUs that are running on this machine. This aggregated data indicates what portion of the machine's resources are being used at any given time. The most useful metric on this chart is the IOWait time measurement, which can indicate performance issues.
- **Load**: The load on the Linux machine indicates the average sum of the number of processes waiting. It includes those processes executing over increments of 1 minute, 5 minutes, and 15 minutes.

Figure 552: Linux Stats performance metrics
CPU usage metrics:

The following selections are available for CPU metrics.

- **Idle**: No threads are running. This statistic indicates the percentage of the machine’s CPUs that are idle.
- **IOWait**: Indicates how long the CPU spends in a waiting state for disk or network I/O.
- **Nice**: View threads with a modified (lower) scheduling priority that configures them to be run when time permits.
- **System**: Displays the percentage of a single CPU that is being used to run system threads.
- **User**: Displays threads initiated by the application (the instance and the database).

Log activity performance graphs

The charts of logging activity display the mean error log count and the mean number of logs created per minute within the configured time period.

The error logging rate is the most important metric in this group.

![Graphs of error and log activity](image.png)

Figure 553: Logging performance metrics
MySQL overview graphs

The graphs in the MySQL Overview page are an aggregate view of all traffic going to your database server.

These graphs are distinct from the charts under the Database section, which monitor only queries originating at your particular application instance.
• **mySQL Statements**: Displays all the database activity - deletes, inserts, selects, and updates.
• **mySQL Threads**: Measures the number of connections to the database and how many active threads are running on the database.
• **mySQL Statement - writes**: Displays only the database write actions, including deletes, inserts, and updates. The select action represents a large portion of the activity, and removing this view makes it easier to analyze the write actions.
• **mySQL Table Locks waited**: Displays the number of table locks per second over the configured time period. This metric is important to monitor. The more table locks that exist, the lower the performance.

Node performance metrics

Node Metrics displays a set of baseline performance and throughput metrics for a particular application instance and/or cluster node.

Most customers are not running in a clustered configuration, but you still have node data because the system treats those instances as a cluster with only one node.

Customers who have multiple cluster nodes are able to view different metrics for each node on their cluster by changing the CI in the graph control.
Figure 555: Node performance metrics

- **Response Time**
  - Graph showing response time (ms) over time.
  - Metrics include Max, Median, and Min.

- **Events Logged**
  - Graph showing events per minute over time.
  - Metric: logs_mean.

- **Events Processed**
  - Graph showing events processed per minute over time.
  - Metric: processed_mean.

- **Transactions**
  - Graph showing transactions per minute over time.
  - Metric: Count.
• **Response Time**: Displays in milliseconds the maximum, median, and minimum response time for database queries by the selected node.

• **Events Logged**: Shows the mean number of the events queued and added to the event log in the selected time period.

• **Events Processed**: Shows the mean number of the events processed during the selected time period.

• **Transactions**: Displays the total number of database transactions per minute by the selected node.

• **Semaphore Use**: Shows the number of semaphores in use by the selected node. Semaphores control the number of user transactions that can be run in parallel. The platform manages semaphores, and they require no customer administration. The semaphore graph is used only by ServiceNow Technical Support for troubleshooting.

• **DB Connection Use**: View the maximum, median, and minimum number of database connections in use by the selected node.

**ServiceNow servlet performance metrics**

Each ServiceNow instance has a servlet, and you can monitor its performance.
Figure 5: Servlet performance metrics

- **Sessions**
  - Bosdemo01.service-now.com:demosw_16245 - 2 hours
  - Active Sessions

- **Transactions**
  - Bosdemo01.service-now.com:demosw_16245 - 2 hours
  - Count

- **Response Time**
  - Bosdemo01.service-now.com:demosw_16245 - 2 hours
  - Max, Median, Min

- **Java Memory**
  - Bosdemo01.service-now.com:demosw_16245 - 2 hours
  - In Use, Max, Total
• **Sessions**: The session data graph can display existing sessions, including those sessions initiated by the MID Server and external integrations.

• **Session Wait Queue**: Displays the number of transactions that are waiting on another transaction to finish because they share the HTTP session. Waiting sessions occur when a user submits a duplicate request before the prior request completes. Items on this list are suitable candidates for creating a transaction quota or for adding to the cancellation white list.

• **Transactions**: Displays all transactions initiated by users, the MID Server, and external integrations.

• **Response Time**: Displays the interval (in milliseconds) between the time the instance receives a transaction and the time the instance responds.

• **Java Memory**: Records memory usage and indicates when the instance is running out of memory. The Java Memory graph is a useful problem indicator.

• **CPU Usage**: Shows aggregated CPU usage for all the instances on the machine. This information is used by ServiceNow Technical Support to troubleshoot performance issues.

• **Scheduler**: Displays all scheduler activity for the selected instance, including Discovery probes. You can determine the backlog of scheduled jobs in the queue for a particular time period. You can then compare that against the rate at which the jobs are being processed during the same period.

### System diagnostics

Administrators can use the System Diagnostics page to look for root causes of platform performance issues.

This page provides an overview of useful diagnostic information about a running instance and cluster nodes.

Navigate to **System Diagnostics > Diagnostics Page**.
### System Diagnostics

#### Cluster Nodes Status

<table>
<thead>
<tr>
<th>Name</th>
<th>app160021.iad4.service-now.com:demoniightyheiski001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>online</td>
</tr>
<tr>
<td>Logged in users</td>
<td>26</td>
</tr>
<tr>
<td>Last reported</td>
<td>0 seconds</td>
</tr>
<tr>
<td>Last reported (seconds ago)</td>
<td>0</td>
</tr>
<tr>
<td>JVM UP time</td>
<td>13 hours 1 minute</td>
</tr>
<tr>
<td>JVM CPU time</td>
<td>3 hours 8 minutes</td>
</tr>
<tr>
<td>Scheduler running</td>
<td>true</td>
</tr>
<tr>
<td>Scheduler queue length</td>
<td>0</td>
</tr>
<tr>
<td><strong>Memory (MB)</strong></td>
<td>813.0 of 1080.0</td>
</tr>
<tr>
<td>JVM Classes</td>
<td>27020.0 toasted, 1503.0 unloaded</td>
</tr>
<tr>
<td>Transactions</td>
<td>82905</td>
</tr>
<tr>
<td>Errors</td>
<td>218</td>
</tr>
<tr>
<td>GC.ConcurrentMarkSweep.Count</td>
<td>329 (2 per 5 minutes)</td>
</tr>
<tr>
<td>GC.ConcurrentMarkSweep.TotalTime</td>
<td>20228.85 (4 minutes)</td>
</tr>
<tr>
<td>GC.ConcurrentMarkSweep.ArgTime</td>
<td>0.88840425</td>
</tr>
<tr>
<td>GC.PanNow.Count</td>
<td>7425 (47 per 5 minutes)</td>
</tr>
<tr>
<td>GC.PanNow.TotalTime</td>
<td>144483 (2 minutes)</td>
</tr>
<tr>
<td>GC.PanNow.AvgTime</td>
<td>0.019463676</td>
</tr>
</tbody>
</table>

#### System overview

| Emails (recv) last 60 minutes                   | 0                                                   |
| Emails (sent) last 60 minutes                  | 0                                                   |
| Events pending                                 | 1                                                   |
| Log entries last 60 minutes                    | 409                                                 |
| POP3 Status                                    | ?                                                   |
| SMTP Status                                    | There is currently no active SMTP account configured, cannot send email |

#### Database overview

| URL                                          | jdbc:mysql://db160144.iad4.service-now.com:3441/af5b33bf177b50a5 |
| Name                                         | demoniightyheiski_1                                    |
| Driver                                       |                                                      |
| Type                                         |                                                      |
| Version                                      | 5.6.16-log                                            |
| JDBC                                         | 1.1                                                  |
Cluster Nodes

If you're running a multiple-node cluster, it's useful to know the status of each of the nodes. The nodes check in every 30 seconds and display their current status in the [sys_cluster_state] table. Navigate to System Cluster > Node States.

Memory

The Cluster Node Status shows used and total memory for each node. Memory use fluctuates, and it is not uncommon for it to go up to beyond 95% usage before memory garbage collection reduces it back to normal levels.

Transactions

If you're looking for information about what a user was doing at a particular time, the transaction log is useful. The transaction log lists all transactions, the user who performed it, and the URL the user was going to. Navigate to System Logs > Transaction Log to see the information.

Email

The email log can help you determine whether emails are being sent, the recipient, and the format of the message. The email log also shows all inbound email. Navigate to System Logs > Email Log.

Configuring System Diagnostics Properties

System Diagnostics properties let you define when items appear red on the page. Navigate to System Properties > System Diagnostics.
System Diagnostics

Please edit your changes and press Save

Customization Properties for System Diagnostics

Auto refresh the system_diagnostics_page after this many seconds:
60

Condition used to flag Emails Received value:

Condition used to flag Emails Sent value:

Condition used to flag Pending Events value:
> 1000

Condition used to flag number of log entries in last 24 hours value:

Condition used to flag a node’s Last Reporting Time value:
> 180

Condition used to flag a node’s Scheduler Is Running value.
!= 'true'

Condition used to flag a node’s Schedule Queue Length value.
> 10

Condition used to flag a node’s Status value:
!= 'online'

Condition used to flag POP3 Status value:
indexOf("Connected") == 1

Condition used to flag SMTP Status value:
Table rotation

Table rotation preserves instance performance and averts risk associated with querying growing data sets by using the External Communication Channel (ECC) Queue and the `sys_created_on` field to separate data sets into multiple tables based on date.

Note: Deployment of this plugin should be executed in partnership with a ServiceNow representative.

Functionality

The administrator specifies the time parameter (duration) of the process and the number of tables (rotations) within. After the rotation writes the last table in a rotation, the rotation overwrites the first table in the rotation. Please contact ServiceNow Technical Support before applying table rotation to a custom table.

Examples:

- The query `Records created between 2015/12/10 08:49 and 2015/12/09 07/34 where topic=SystemCommand` is translated to a SQL query on a single table, because the clause on `sys_created_on` targets a single shard.
- The query `Records updated between 2015/12/10 08:49 and 2015/12/09 07/34 where topic=SystemCommand, or without a date range, needs to target all shards and therefore is translated as a union query on all shards.

Advantages

- Allows deletion of old data without affecting current data (for example, to drop or truncate a table).
- Ensures tables only grow to a reasonable size.
- Reduces working set of data when date is known for query.

Disadvantages

Queries that do not use the table rotation date (for example, by using the `sys_created_on` field), force an inefficient union query to query time ranges that span multiple tables and can be extremely slow if the number of sub-tables is large.

To improve performance, it is recommended that the query includes a window of created dates.

You can use table rotation for sequentially-written tables or for insert-only tables. You cannot use table rotation for `sys_import` tables or tables that extend the Task `[task]` table.

Apply table rotation

You can apply table rotation by specifying the time parameter (duration) of the process and the number of tables (rotations) within.
Role required: admin

**Note:** Deployment of this plugin should be executed in partnership with a ServiceNow representative.

1. Navigate to **System Definition > Table Rotations**.
2. Click **New**, or select the table rotation group to modify.
3. Set the table rotation fields.

**Table 484: Table rotation group fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the table to which the action is to be applied.</td>
</tr>
<tr>
<td>Duration</td>
<td>Set the overall duration for the action.</td>
</tr>
<tr>
<td>Initialized</td>
<td>Sets table rotation as active (true) or inactive (false). [This must be checked (true) for the rotation or extension to be active.]</td>
</tr>
<tr>
<td>Type</td>
<td>Choose either Rotation or Extension.</td>
</tr>
<tr>
<td>Rotations</td>
<td>Sets the number of tables to be maintained through the duration.</td>
</tr>
<tr>
<td>Clean base rotation</td>
<td>Sets the date to clean (truncate) the base table.</td>
</tr>
</tbody>
</table>

4. Click **Submit** or **Update**.

**Note:** Deleting a rotation deletes the additional tables and all the data. Do not delete the rotation if the data is needed.

When you define a new rotation, a schedule is created and new data is subsequently written to one of the tables in the rotation group. The group includes the original table plus a number of additional tables.
## Table Rotation Group - syslog

**Name**: syslog  
**Duration**: Days 7, Hours 00  
**Type**: Rotation  
**Rotations**:  
**Clean base rotation**: 2017-01-12 15:14:25

### Related Links
**Synchronize Shards**

### Table Rotation Schedule

<table>
<thead>
<tr>
<th>Name = syslog</th>
<th>Offline</th>
<th>Table name</th>
<th>Valid from</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>Log Entry [syslog0007]</td>
<td>2017-01-04 03:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog0006]</td>
<td>2016-12-28 03:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog0000]</td>
<td>2016-11-16 03:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog0005]</td>
<td>2016-12-21 03:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog0003]</td>
<td>2016-12-07 03:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog0001]</td>
<td>2016-11-23 03:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog0004]</td>
<td>2016-12-14 03:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog]</td>
<td>1969-12-31 16:00</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>Log Entry [syslog0002]</td>
<td>2016-11-30 03:00</td>
<td></td>
</tr>
</tbody>
</table>

### Actions on selected rows
Task table flattening

Within the ServiceNow platform, a logical table, such as the task list, represents one or more physical tables in a relational database.

Hierarchically related tables, such as Task [task], Incident [incident], Problem [problem], and Change [change_request], can be stored in one of two different extension models. The Table per class extension model stores hierarchically related tables as separate physical tables. The Table per hierarchy extension model stores hierarchically related logical tables as a single, flat, physical table and provides a performance improvement over multiple, joined tables.

The ServiceNow platform uses the table per hierarchy extension model for the Task table on MySQL databases. Other tables use the table per class extension model because there is no performance benefit to flattening them. To use table per hierarchy on an Oracle database, contact Technical Support.

Advantages of flattening a table hierarchy

The ServiceNow platform uses table joins to associate hierarchically related tables in the Table per class model. Queries against any child table, such as Incident, must join the child table to the parent table, such as Task. These table joins cause a performance bottleneck when performing queries. This bottleneck is especially noticeable when a query accesses fields from multiple child tables, such as when querying both Incident and Problem table values.

A flat table hierarchy allows the ServiceNow platform to access the data across these logical tables without first joining them because the data is stored in the same physical table.

Flattening a table hierarchy does not affect the appearance or functionality of tables as they appear in the ServiceNow platform. All table functionality, including database views, remains unchanged by flattening the table hierarchy.

View a table hierarchy and extension model

The extension model used by a table is not immediately apparent. While a hierarchy can use a single physical table, the platform displays tables as if each logical table has a unique physical table. For example, when specifying a table for a workflow, you can select Change [change_request] or Incident [incident] even though the parent table, Task [task], uses a single physical table.

Role required: admin

Administrators can view the status of flattened table hierarchies, but cannot flatten additional hierarchies. You can configure the form to add the Extension model field if necessary.

1. Navigate to System Definition > Tables.
2. Select a table record.
3. Review the Extension model field value, which indicates whether the table hierarchy uses multiple unique tables or a single flat table.

   • **None**: Indicates that the table uses the table per class model. Defines a unique physical table per logical table in a hierarchy. For example, there are separate physical tables for Task [task], Incident [incident], Problem [problem], and Change [change_request] on the MySQL database.

   • **Table per hierarchy**: Defines a single physical table per logical table hierarchy. For example, there is a single Task [task] table on the MySQL database that contains all the Task, Incident, Problem and Change records. This single physical table is represented as separate logical tables.
Add a module to test connection speed

A connection test can indicate the connection speed between your computer and your ServiceNow instance. A connection speed test (/connection_test.do) is available as a UI page.

Role required: admin

![Connection Test](image)

Connection test is used to determine the type of connect between the end computer and your Service-now.com instance.

To start the connection test, press this button:

Start Connection Test

Average time: 594ms (per 170k)
Estimated connection: T1/Cable (average download was 282K/s)

Figure 559: Connection Test page

1. Perform the appropriate action for your version of the UI:

| UI16          | 1. Navigate to **System Definition > Application Menus**.  
|              | 2. Open the application menu to which you want to add the module, for example, **System Diagnostics**. |
| UI15 or UI11 | Right-click an application menu, such as **System Diagnostics**, and select **Edit Application Menu** or click the pencil icon. |

2. Click **New** in the list of modules.
3. Complete the Module form using the following values:
   - **Title**: Unique name such as **Connection Test**.
   - **Link type**: **URL (from Arguments)**
   - **Arguments**: connection_test.do
4. Click **Submit**.

**This ServiceNow instance**

Performance graph statistics are attached to the **This ServiceNow instance** business service. These statistics are the ones you can see on the Admin homepage when you add a Performance Graph Controls widget, and then add one or more performance graphs.

Do not delete this business service, or else the instance becomes unable to track its own performance.
Figure 560: This ServiceNow Instance
Transaction quotas

Transaction quotas allow you to define a quota policy for different types of transactions. A transaction quota cancels any transaction in violation of the policy and notifies the user of the cancellation.

The Transaction Quotas plugin is active by default on all new and upgraded instances.

![Image of transaction canceled](image1)

**Figure 561: Transaction canceled**

The transaction quota also writes the cancellation message to the log file as a warning.

![Image of log entry](image2)

**Figure 562: Transaction cancelled log**

Typically, administrators set transaction quotas to prevent poorly performing queries and scripts from consuming system resources. This ensures that no transaction consumes enough resources to prevent other transactions from running. Administrators can also view cancellation log messages to identify transactions that might consume excessive resources.
How transaction quotas work

Transactions use the Quota Manager, which is a background thread that cancels transactions. The Quota Manager performs the following actions.

1. Obtains a list of active transactions, similar to the list under User Administration > Active Transactions.

2. Cycles through each transaction, and matches it to a quota rule if it does not have one assigned to it. If a transaction changes or there is a new quota rule, the Quota Manager reprocesses the transaction.

3. Cancels the transaction if it has been running longer than the specified quota maximum.

4. Logs the running transactions.

5. Sleeps until the next heartbeat, which is controlled by the glide.quota.manager.heartbeat system property.

Transaction cancellation examples

Transactions can be canceled for more than one reason. Look for the following indicators.

- maximum execution time exceeded: This message appears when the glide.quota.manager threads cancels the transaction.
- canceled by other transaction: This message appears when the transaction was canceled by the session thread that initially issued it, and not by the glide.quota.manager thread.
- canceled by user request: This message appears when the user clicked the red X button to cancel the running transaction.

If you are looking for transactions that were canceled by glide.quota.manager, Search for the [Message] [Contains] [maximum execution time exceeded] combination. For example:

Cancelling transaction /home.do - Default-thread-11.0 (maximum execution time exceeded): Thread Default-thread-11.0 (Default-thread-11.0, F530DD111B11111111FC031767DA158E), after 30000ms

Transaction quota properties

The administrator can add the following system properties to manage transaction quotas.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.quota.manager.heartbeat</td>
<td>The number of seconds between the start of each Quota Manager heartbeat. This value determines how often the Quota Manager checks for transactions that exceed a quota and how often it writes status in the log file.</td>
</tr>
</tbody>
</table>
|                              | Type: Integer
|                              | Default value: 1
<p>|                              | Location: Add to the [sys_properties] table                                                                                                    |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.quota.manager.minimum_transaction_time</td>
<td>The minimum number of seconds a transaction must run before the Quota Manager matches it to a transaction quota. ServiceNow recommends setting this value to at least 1 second to avoid performance issues. For optimal performance, set this value to the value of your most restrictive quota. For example, if your most restrictive quota cancels transactions longer than 1 minute, set the minimum transaction time to 60 seconds.</td>
</tr>
<tr>
<td></td>
<td>Type: Integer</td>
</tr>
<tr>
<td>glide.quota.manager.debug</td>
<td>Controls whether to display (true) or hide (false) additional debugging information related to the Quota Manager. Debugging information includes running transactions, canceled transactions, and what quotas are matched to transactions.</td>
</tr>
</tbody>
</table>

Enable transaction quota debugging
You can enable the logging of transaction quota debugging information by enabling the system property.

Role required: admin

1. Add the `glide.quota.manager.debug` system property, and set the value to true.
2. Navigate to System Diagnostics > Session Debug > Debug Quotas.
   The Debug Quota script runs to enable debugging.

After setting the above system property to true, go to your system log to see messages from the Quota Manager.

Default quota rules
Various transaction quota rules are available in the base system.

- **Fix Script Processor**: Allows the fix script processor to run for four hours.
- **Presence**: Cancels presence requests quickly when the system is busy.
- **REST Import Set API request timeout**: Prevents inbound REST Import Set API transactions from running for longer than 60 seconds.
- **REST Table API request timeout**: Prevents inbound REST Table API transactions from running for longer than 60 seconds.
- **AMB Transactions**: Cancels AMB transactions lasting longer than the specified maximum duration. Applies to all AMB transaction types: message send and message receive.
- **Reference Completer**: Stops the reference completer transaction after five seconds.
- **REST Aggregate API request timeout**: Prevents inbound REST Aggregate API transactions from running for longer than 60 seconds.
- **Homepage Widgets**: Prevents all homepage widgets from running longer than 30 seconds.
• **REST Attachment API request timeout**: Prevents inbound REST Import Set API transactions from running for longer than 60 seconds.

• **UA Count Persistor Quota**: Prevents UA count persistor scheduled jobs from running for more than one hour.

• **UI Transactions**: Cancels UI transactions that are two seconds away from the 5-minute server disconnect. ServiceNow datacenter load balancers display an internal server error (HTTP Error 500) to users after five minutes. This quota rule returns a transaction canceled page instead of an error. This rule includes an exception to prevent the cancellation of background scripts.

### Normal transaction activities

Before setting transaction quotas, navigate to **User Administration > Active Transactions** to review the normal transaction activities for your instance. You can **view and terminate long-running transactions** if necessary. Over time, you can determine the normal transaction load for your instance and set your quotas to match these norms.

**Configure a transaction quota rule**

Transaction quota rules allow you to define a quota policy for different types of transactions.

Role required: admin

Transaction quota rules allow you to specify the following conditions.

- The conditions under which the policy applies.
- The order in which transaction quotas apply.
- The maximum duration of a transaction before the quota manager cancels it.

**Warning**: Setting transaction quotas too low can severely impact your users and prevent normal instance operations. Test your transaction quotas rigorously before implementing them in production.

1. If necessary, activate the Transaction Quotas plugin.
2. Navigate to **System Definition > Quota Rules**.
3. Click New.
4. Complete the form.

### Table 486: Transaction quota rule fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name to identify the rule.</td>
</tr>
<tr>
<td>Active</td>
<td>A check box that determines if this rule is active (selected).</td>
</tr>
<tr>
<td>Maximum Duration</td>
<td>The number of seconds a transaction has to complete before the quota cancels the transaction.</td>
</tr>
<tr>
<td>Maximum Business Rules</td>
<td>The number of business rules executions allowed.</td>
</tr>
<tr>
<td>Maximum Database Time</td>
<td>The total number of seconds for all SQL requests.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximum SQL Statement Time</td>
<td>The number of seconds a SQL statement can run.</td>
</tr>
<tr>
<td>Maximum Outbound Requests</td>
<td>The number of outbound HTTPs requests allowed.</td>
</tr>
<tr>
<td>Order</td>
<td>A number that represents the priority of the quota transaction in relation to other quotas. The transaction quota with the lowest order value and matching conditions determines the quota policy that is applied.</td>
</tr>
<tr>
<td>Maximum Events</td>
<td>The number of sysevent inserts allowed.</td>
</tr>
<tr>
<td>Maximum Jobs</td>
<td>The number of sys_trigger inserts allowed.</td>
</tr>
<tr>
<td>Maximum SQL Queries</td>
<td>The number of SQL queries allowed.</td>
</tr>
<tr>
<td>Maximum Outbound Request Duration</td>
<td>The number of seconds for an outbound HTTP request.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the transaction quota rule.</td>
</tr>
</tbody>
</table>

5. Set a condition to specify when the transaction quota should take effect.

The transaction quota rule condition builder displays conditions that are only applicable to transactions as listed in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>The URL of the page.</td>
</tr>
<tr>
<td>Thread Name</td>
<td>The name of the execution thread that is running the transaction. Foreground threads have names like http-bio-8080-exec-1. Background threads can have various names, and are useful where the URL is not sufficiently descriptive.</td>
</tr>
<tr>
<td>Foreground</td>
<td>Whether the transaction was completed in the foreground or background:</td>
</tr>
<tr>
<td></td>
<td>• <strong>True</strong>: a foreground transaction.</td>
</tr>
<tr>
<td></td>
<td>• <strong>False</strong>: a background transaction.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>The type of transaction:</td>
</tr>
<tr>
<td></td>
<td>• <strong>List</strong>: Any list transaction, such as incident_list.do.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Form</strong>: All forms, UI pages, CMS pages, and so on.</td>
</tr>
<tr>
<td></td>
<td>• <strong>XMLHttp</strong>: Transactions that run through GlideAjax, where the URL is xmlhttp.do.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Report</strong>: The page sys_report_template.do.</td>
</tr>
<tr>
<td></td>
<td>• <strong>SOAP</strong>: SOAP transactions.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Export</strong>: When a list is exported in a format such as XML or Excel.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Scheduler</strong>: When a scheduled job is performed.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text Search</strong>: The text search transaction or any of its related operations.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Other</strong>: Any type that does not meet the qualities of another type.</td>
</tr>
</tbody>
</table>

**Note**: Match the transaction quota **Type** to the transaction type listed in the Quota Manager log entry for the transaction.

<table>
<thead>
<tr>
<th>User</th>
<th>The user performing the transaction.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homepage</td>
<td>The homepage that is running. This condition is populated only if the URL is home.do.</td>
</tr>
<tr>
<td>Homepage Widget</td>
<td>The homepage widget that is rendered in the transaction. When a homepage is rendered initially, each homepage widget is rendered in parallel.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Miscellaneous attributes that are related to the transaction.</td>
</tr>
</tbody>
</table>

6. Click **Submit**.

**View a canceled transaction**

The Quota Manager logs each canceled transaction as a warning message in the system log.

Role required: admin

Use the following procedure to search for transaction warnings in the system log.

1. Navigate to **System Logs > System Log > Warnings**.
2. Edit the filter to add the condition **[Message] [starts with] [Cancel]**.
3. Click **Run**.
Example system log messages
An example of system log messages for transactions.

- At every heartbeat interval, which is one second by default, the Quota Manager prints the running transactions:

```
2012-02-13 12:34:08 (096) glide.quota.manager SYSTEM URL= /incident_list.do?
sysparm_userpref_module=b55fbec4c0a80090088e83d7ff500de&active=true&sysparm_query=active=true^EQ,
THREAD= http-bio-8080-exec-3, FG= true, TYPE= 1, STATE= 2, USER= null,
TIME= 8,807, MEM= 0, ATTRIBUTES= {}
```

- Every time the Quota Manager matches a quota to a transaction, it prints a message similar to the following example:

```
2012-02-13 13:25:31 (900) glide.quota.manager SYSTEM QuotaFinder: Assigning quota "TEST PROBLEM FORM" with filter:
type=form^urlLIKEsys_id=46fb9e31a9fe198101492060c2a4f8cb^EQ to transaction: URL= /problem.do?sys_id=46fb9e31a9fe198101492060c2a4f8cb,
THREAD= http-bio-8080-exec-1,
FG= true, TYPE= 1, STATE= 4, USER= null, TIME= 1,121, MEM= 0, ATTRIBUTES= {}
```

- Every time the Quota Manager cancels a transaction, it prints a message similar to the following example:

```
2012-02-13 13:25:33 (930) glide.quota.manager SYSTEM WARNING *** WARNING ***
Transaction: Cancelling transaction /problem.do (maximum execution time exceeded):
Thread http-bio-8080-exec-1
```

Modify the transaction cancellation page
The Quota Manager uses a UI page to control the contents of the transaction cancellation message.

Role required: admin

Knowledge of Apache Jelly is highly recommended when modifying the UI page. See Extensions to Jelly syntax on page 3937 for more information.

1. Navigate to System UI > UI Pages.
2. Open the UI page with the name transaction_canceled_quota.
3. In the HTML field, add or modify the new cancellation message.

The following table describes available variables.
Table 487: Cancellation message variables

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getURL()</td>
<td>Returns the URL for the transaction.</td>
</tr>
<tr>
<td>getReason()</td>
<td>Returns the reason for page cancellation, localized to the user’s language.</td>
</tr>
<tr>
<td>getQuotaId()</td>
<td>Returns the sys_id of the quota rule.</td>
</tr>
<tr>
<td>getQuotaName()</td>
<td>Returns the name of the quota rule.</td>
</tr>
<tr>
<td>getRunTime()</td>
<td>Returns the total run time for the page up to this point.</td>
</tr>
<tr>
<td>getType()</td>
<td>Returns the type of transaction, such as form, list, report, other.</td>
</tr>
<tr>
<td>getUser()</td>
<td>Returns the sys_id of the user.</td>
</tr>
<tr>
<td>getHomepage()</td>
<td>Returns the sys_id of the homepage.</td>
</tr>
<tr>
<td>getHomepageWidget()</td>
<td>Returns the name of the homepage widget.</td>
</tr>
</tbody>
</table>

4. Click **Update**.

Use table extension

Table extension is a data archiving technique used to preserve instance performance and avert risk associated with querying growing data sets. To do this, table extension separates data sets into multiple tables based on date.

Role required: admin

This example outlines how to use table extension.

**Note:** Deployment of this plugin should be executed in partnership with a ServiceNow representative.

The administrator of the process is given the ability to specify time parameter (duration) of the process. Although the administrator can also specify the number of tables (rotations), this is not the best practice. After the last table in a rotation is written, new tables are added and old tables are archived. Using table extension, tables are never overwritten.

An advantage of table extension is to partition data across tables. It also allows you to archive data while ensuring that tables stay reasonably-sized. The working set of data is reduced when a date is known for the query.

The disadvantage is that table extension equires a union query when you query for a time range that spans multiple tables. Union queries are less efficient than queries against a single table.

A good practice is to use table extension when you have sequentially-written tables or insert-only type tables (there are exceptions to this parameter). Table extension is also useful in tables where data is needed for long periods of time.

The following example describes how to set table extension for the [sys_audit] table.

1. Navigate to **System Definition > Table Rotations**.
2. Click **New**.
3. Enter the following information.
<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the table to apply the action. In this example, the table is [sys_audit].</td>
</tr>
<tr>
<td>Duration</td>
<td>Enter the overall duration for the action. 30 days is the duration in this example.</td>
</tr>
<tr>
<td>Type</td>
<td>Select Extension.</td>
</tr>
</tbody>
</table>

4. Click Submit.

Content Management System

The Content Management System (CMS) is a ServiceNow application that primarily enables users to create a custom interface for the ServiceNow platform and ServiceNow applications.

The CMS application is powerful and flexible, so customers have used it for a wide variety of projects from creating entire websites to integrating with other products. The Content Management application is active by default.

This video provides an overview of the CMS application.

A CMS typically requires a systems administrator or a web developer to set up and add features. Non-technical users can use the application as a website maintenance tool for updates. For more information, see CMS Planning.

Following are several CMS project ideas:

- Design a company-wide service catalog that offers a collection of services.
- Present a customized UI for a knowledge base.
- Create customized login pages, search pages, views of lists, tables, charts, or graphs.
- Design a complete website.
- Integrate ServiceNow with other company applications.
• Build a tailored self-service portal for end users that is in compliance with a corporate style guide.

**Example CMS sites**

There are two common interface approaches within the ServiceNow community:

• An image and text-based interface similar to Amazon.com
• A search-based interface similar to Google

Both approaches have been used successfully. The approach you select depends on the needs of the people using the data and how easy it is to train them. While the two design philosophies are different, both approaches share the common goal of UI simplicity.

**Content Management System design**

Before building a website in CMS, it is important to have a good understanding of what to build and who the audience is.

A high volume of content can heavily influence the look and feel of the site and the site hierarchy. When determining the content, design for ease of maintenance for the people who take care of the system.

Review website design prerequisites to help you set expectations, scope deliverables, and define reasonable time lines. Consider both the planning and execution of site design, to understand how sites are built, and to provide a working overview of the CMS.

Timing is important when considering the addition of content management. Successful deployments of the content management system usually take place after phase 1 processes (for example, Incident, Problem, Change, Catalog, and Knowledge) are in place. This is especially true if the team has limited website design experience. Waiting until phase 2 of deployment gives administrators time to work in ServiceNow and to understand how the organization uses the system and what business needs it meets.

Also consider the maturity level of data in the ServiceNow system. Depending on the ITIL processes used, content management is only useful and effective once the data within ServiceNow is established. Ensure that hierarchies, tasks, and workflows are well-defined. For example, before creating a catalog interface, confirm that the service catalog has been in place for some time, has been used, and contains data. The same is true for a knowledge management interface, particularly when high ratings or view counts define article placement on the page.

<table>
<thead>
<tr>
<th>Prerequisite</th>
<th>Ensures that</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organize the content</strong></td>
<td>System content and CMS content is complete.</td>
</tr>
<tr>
<td><strong>Identify team members</strong></td>
<td>The process is not delayed by a lack of resources.</td>
</tr>
<tr>
<td><strong>Prepare the data within the instance</strong></td>
<td>The instance has all the data the CMS presents.</td>
</tr>
<tr>
<td><strong>Leverage the corporate style guide</strong></td>
<td>The CMS site conforms to the corporate look and feel.</td>
</tr>
<tr>
<td><strong>Use prototypes and rapid web design</strong></td>
<td>The structure of the CMS site is clear and planned out, either on paper or an electronic file.</td>
</tr>
</tbody>
</table>
Content organization

Organize your content before you begin building your website.

A site created in CMS relies on two different types of content.

- CMS site information, such as site pages, images, and menus
- System information, such as knowledge base articles and catalog items

CMS content

Begin by listing all the content you want to host on the CMS pages. Examples include help pages, My Requests, My Approvals, and specific catalog items. Think about current solutions that you can implement immediately, and note ideas for future implementation phases.

Within CMS, you group pages to define the top-down menu structure. You establish a home or starting page, and other pages in the site reference the home page in the Parent Page reference field.

There are several ways to group, such as by audience or the purpose of the website. After listing the content to host, group it logically and identify a common name for each group, as shown in the following examples.

Table 489: IT environment groups

<table>
<thead>
<tr>
<th>Content built for</th>
<th>Common name</th>
</tr>
</thead>
<tbody>
<tr>
<td>End user</td>
<td>End-User Page</td>
</tr>
<tr>
<td>IT professional</td>
<td>IT Professional Page</td>
</tr>
</tbody>
</table>

Table 490: General groups

<table>
<thead>
<tr>
<th>Purpose of site</th>
<th>Common name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports</td>
<td>Reporting Page</td>
</tr>
<tr>
<td>Help and knowledge</td>
<td>Knowledge page</td>
</tr>
</tbody>
</table>

System content

Organize the content so the interface is easy to navigate and understandable to the user. Determine the organization based on the data that you are leveraging, both in the CMS (using sites, parent pages, pages, and navigational menus) and throughout the rest of the system. For example, within the catalog you have “category,” and in the knowledge base you have “category” and “subcategory.” You can use these hierarchies with filtered lists for good search results.

Organizing CMS content logically is important for long-term maintenance of the site, however, the data typically comes from other ServiceNow applications. Communicate with the administrators for these applications, such as the knowledge base, service catalog, and business service portfolio. Work with them to offer the application data appropriately through the CMS pages you create. For example, the team that created the ServiceNow corporate website in CMS began by evaluating the naming conventions used in the corporate knowledge base.
Branding elements

Branding refers to the logo, name, colors, and symbols that identify an organization. It imposes consistency in design and use of terms. Your marketing department defines branding elements and can provide them to you as you plan your CMS pages. Consider how to incorporate the following branding elements.

- Logos
- Color palette
- Tag line
- Trademarked elements
- Graphics

Site design

During planning, consider providing a core set of features with a standard appearance throughout the site. The following web design elements are often used to create a consistent look.

- Page templates
- Navigation schemes
- Header
- Breadcrumbs
- Footer
- Forms

ServiceNow features

Analyze and organize the following ServiceNow features in your instance if you plan on using any of them with CMS pages.

- Account settings
- Email
- Workflow approvals
- Filters

Team member identification

Before starting to build a CMS, it is important to identify the members of the team and their responsibilities. Identify these resources before beginning to build the CMS website or any pages:

- The person responsible for gathering corporate style design guidelines, defining the copy (written terminology and content for the site), and defining the site flow.
- The CMS webmaster who works with the first resource, executes the designs, and makes the site work. The webmaster needs the following skills:
  - Basic ServiceNow administration skills
  - HTML
  - CSS
  - Graphic design
  - Web design
After you identify the project team members, establish who is responsible for the tasks that must be completed as the CMS website is build.

- Determine who owns each page.
- Set a page update schedule so owners do not overwrite each other.
- Formalize content management processes, including content review and page updates.

**Data preparation**

It is important that there is enough data in the ServiceNow instance before you begin to build the CMS website.

Ensure that the necessary data and content are available in the instance by taking the following actions.

- Review the ITIL processes that you intend to implement in the CMS. Ensure that the data in the instance is a mature representation of the applications being used, for example, service catalog, knowledge management, and incident management.
- Review the defined hierarchies, such as the categories and subcategories for the knowledge base or service catalog. You use these categories to design the entry page into the application.

**Corporate style guide**

When you build a CMS website, the corporate style guide dictates the look and feel of your website.

A corporate style guide provides detailed information for designing any corporate interface, including corporate websites.

- **Corporate design team**
  Many organizations have a corporate team that designed their website. Contact this team and involve the designers early in the planning. They can provide help and give their approval to the interface you design. Without approval, there is the risk of having to redesign the entire site because it does not adhere to the organizational guidelines.

- **Corporate style guide**
  A corporate style guide takes the guesswork out of designing the CMS website. The example style guide shown is defined down to the pixel. Creating a site with the style guide makes it easy to create clean CSS and HTML. Without a style guide, building the site can take a great deal of time.

- **Design considerations**
  Some modifications to the base design for forms may be necessary. The content area of any CMS design must be no smaller than 860px, or service catalog forms are clipped. The sample style guide entry specifies the content area to be 576px, which clips service catalog forms.
Columns and topics:

1. Navigation (levels 2–4)
2. Main content
3. Info snippets
Prototypes and rapid web design

Many user interface designers use prototypes and rapid web design techniques to visualize the final product before it is developed.

Design revisions are the most time consuming and expensive phase of site design. When the team analyzes and then uses prototypes to create pages, revisions to the published pages are minimal.

Develop the prototype and print it. Review the design with the appropriate team members and annotate the prototype, and annotate what to update on each page.

Define the following elements within the prototype.

- The site map for the entire site.
- A detailed prototype of every intended page, including elements such as links, link destinations, content, page names, and page descriptions.

Content Management navigation

After you design the hierarchical structure of the site, begin planning site navigation.

Consider the following as you plan site navigation.

- Placing navigation elements on every page.
- Locating navigation elements in the same place on every page.
- Using either text or images for navigation.
- Providing visitors with an easy way to understand where they are in the site, for example, with breadcrumbs or a specific color scheme.
- Adding a site map, which is a one-page, hyperlinked, hierarchical outline of the site.
- Providing a link to the home page from every site page, as visitors often enter the site on a page other than the home page.
- Designing navigation to help visitors find information quickly with as few clicks as possible.

In the CMS, navigation menus define your site navigation. Build menus as navigational blocks to create navigation paths. The menu chosen for the task depends on the size and complexity of your site. Available menu options include the following items.

- List menus
- Two types of tab menus
- Vertical, clickable list, usually placed on the left side
- Horizontal blocks
- Vertical blocks
- Super menu (a menu of menus)

System content management

Most of the content in a CMS site is managed in different locations throughout the system.

For example, if you are building a knowledge website, the pages and blocks exist in CMS, but the knowledge articles are authored and managed in the Knowledge application. The same is true for any other type of content you plan to use. It is important to take time to understand the table structure of data to become acquainted with content.

Links to content are typically static, however, take time to look at the document tree and review how field values are formatted for use within the CMS. To understand the information provided, right-click within forms in the platform and select Show XML to view the document tree for the referenced table. To see
the table values for each field, right-click the form label and choose Show - (field name) or Configure Dictionary for reference.

Look at several internet news sites for ideas on how to format dynamic list data and also the full article detail. Research blog sites, shopping sites, and any other site you find easy to use, as layout and usability design can be time-consuming. If you find a site that inspires you, emulate it in your design.

• This New York Times example has two separate list formats.
• The CNN example has several list formats on the page.
• Several different list formats are used on the ServiceNow website.

Knowledge articles - kb_knowledge table

When you right-click and select Show XML on any form within the system, the document tree for the referenced database table becomes reference-able. Review the following selected subset of the document tree so you can acquaint yourself with the content readily available to your site design.

```xml
<?xml version= "1.0" encoding= "utf-8" ?>
<j:jelly trim = "false" xmlns:j = "jelly:core" xmlns:g = "glide" xmlns:j2 = "null" xmlns:g2 = "null" >
    <div class = "cms_knowledge_list customer_success" >
        <g:for_each_record file = "${current}" max = "${jvar_max_entries}" >
            <br />
            <table cellspacing = "0" cellpadding = "0" border = "0" class = "background_transparent" >
                <tr><td class = "cms_knowledge_list_image"
                    <j:if test = "${current.u_logo.getDisplayValue() != ''}" >
                        <div class = "knowledge_article_logo" >
                            <a href = "knowledge.do?sysparm_document_key=kb_knowledge,${current.sys_id}" >
                                <img src = "${current.u_logo.getDisplayValue()}" alt = "${current.text}" width = "110px" />
                            </a>
                        </div>
                    </j:if>
                </td><td width = "100%" ><a href = "knowledge.do?sysparm_document_key=kb_knowledge,${current.sys_id}" target = "_top" >
                    <span class = "kb_learn_more" >
                        $\{current.short_description\}
                    </span>
                </a>
                </td></tr>
                <tr><td width = "100%" colspan = "2" class = "kb_learn_more" >
                    <a href = "knowledge.do?sysparm_document_key=kb_knowledge,${current.sys_id}" >Learn More</a>
                </td></tr>
            </table>
        </g:for_each_record>
    </div>
</j:jelly>
```
Activate the Content Management System

The Content Management System (CMS) application and the Service Management Portal application, which is implemented within Content Management, are automatically installed on new instances. If they are not active on your instance, administrators can activate them.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Installed with Content Management

Several types of components are installed with the Content Management application.

Demo data is available for this feature.

Tables installed with Content Management

Content Management adds the following tables.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Page [content_page]</td>
<td>Provides an inventory list of pages within the system</td>
</tr>
<tr>
<td>Content CSS [content_css]</td>
<td>Stores internal or external CSS for the page</td>
</tr>
<tr>
<td>Content Theme [content_theme]</td>
<td>Made up of a collection of CSS files</td>
</tr>
<tr>
<td>Content Link [content_link]</td>
<td></td>
</tr>
<tr>
<td>Content Page Rule [content_page_rule]</td>
<td></td>
</tr>
<tr>
<td>Content Block Programmatic [content_block_programmatic]</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Content Block</td>
<td>[content_block]</td>
</tr>
<tr>
<td>Content Block Detail</td>
<td>[content_block_detail]</td>
</tr>
<tr>
<td>Content Site</td>
<td>[content_site] Provides and inventory list of sites within the system</td>
</tr>
<tr>
<td>Content Block Static</td>
<td>[content_block_static]</td>
</tr>
<tr>
<td>Content Type Detail</td>
<td>[content_type_detail]</td>
</tr>
<tr>
<td>Content Page Meta</td>
<td>[content_page_meta]</td>
</tr>
<tr>
<td>Content Config</td>
<td>[content_config]</td>
</tr>
<tr>
<td>Content Type</td>
<td>[content_type]</td>
</tr>
<tr>
<td>Content Theme CSS</td>
<td>[content_theme_css]</td>
</tr>
<tr>
<td>Content Block Lists</td>
<td>[content_block_lists]</td>
</tr>
<tr>
<td>Content Block Sized</td>
<td>[content_block Sized]</td>
</tr>
</tbody>
</table>

### Role installed with Content Management

Content Management adds the following role.

<table>
<thead>
<tr>
<th>Role title [name]</th>
<th>Description</th>
<th>Contains roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content administrator [content_admin]</td>
<td>Can read, write, and configure all elements of the Content Management Application</td>
<td>none</td>
</tr>
</tbody>
</table>

### Script includes installed with Content Management

Content Management adds the following script includes.
<table>
<thead>
<tr>
<th>Script include</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSCopyAjax</td>
<td>Support for copying CMS pages</td>
</tr>
<tr>
<td>CMSEntryPage</td>
<td>Creates an entry page</td>
</tr>
<tr>
<td>ContentPageClone</td>
<td>Copy support for CMS pages</td>
</tr>
<tr>
<td>ContentSiteClone</td>
<td>Copy support for CMS sites</td>
</tr>
<tr>
<td>CMSAjax</td>
<td>CMS Ajax utilities</td>
</tr>
</tbody>
</table>

### Client script installed with Content Management

Content Management adds the following client script.

<table>
<thead>
<tr>
<th>Client script</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Color Change</td>
<td>Content Block Header</td>
<td>Changes the background color of the control to what the user selected</td>
</tr>
<tr>
<td></td>
<td>[content_block_header]</td>
<td></td>
</tr>
</tbody>
</table>

### Business rules installed with Content Management

Content Management adds the following business rules.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace Stub</td>
<td>Content Block</td>
<td>Replaces a placeholder widget with a content widget.</td>
</tr>
<tr>
<td></td>
<td>[content_block]</td>
<td></td>
</tr>
<tr>
<td>Unload Page</td>
<td>Page</td>
<td>Unloads content from the page.</td>
</tr>
<tr>
<td></td>
<td>[content_page]</td>
<td></td>
</tr>
<tr>
<td>Bump Parent if Content</td>
<td>Portal</td>
<td>Replace content in the current page.</td>
</tr>
<tr>
<td></td>
<td>[sys_portal]</td>
<td></td>
</tr>
<tr>
<td>Remove Block from Pages</td>
<td>Content Block</td>
<td>Removes a block from pages where it is placed.</td>
</tr>
<tr>
<td></td>
<td>[content_block]</td>
<td></td>
</tr>
<tr>
<td>Validate Suffix</td>
<td>Content Page</td>
<td>Validates a suffix so that a page name or a page suffix cannot match a table name.</td>
</tr>
<tr>
<td></td>
<td>[content_page]</td>
<td></td>
</tr>
</tbody>
</table>

### Configure Content Management sites

Planning a CMS site involves obtaining resources, communicating with others about design, and gathering content.

Role required: content_admin or admin

The following steps are a high-level overview of how to set up a site.

1. Plan and design your content.
2. Create a site.
The site is the container that holds all the content. To simplify your process, you can copy an existing site and edit its components. For more information on creating a site, see Create a site on page 2052. For more information on copying a site, see Copy a site on page 2056.

3. Add pages to the site.
   Pages contain blocks of information for the site. For more information, see Create a content page on page 2058.

4. Create content blocks to customize the layout, headers, menu navigation, lists, and static and dynamic content.
   Content blocks are chunks of HTML that make up the content page. There are various content block types available to help with your customization. For more information, see Content blocks on page 2072.

5. Add style to your site using themes, style sheets, and frames.
   Content sites and content pages do not reference style sheets directly. Configure them using the Themes or Style Sheets options. For more information, see Style in Content Management on page 2105.

6. Test the site.
   After you create or modify a site, test the site to ensure that content displays properly and all links work correctly. For more information, see Test the site.

Content sites

A content site is a group of related content pages that have the same basic theme, layout, and URL suffix.

Content sites are made up of a series of basic building blocks. The Employee Self-Service site is an out-of-box sample site that is included in the CMS activation. It provides existing, working examples or each CMS component, which you can use like a template to build your own site.
Create a site

A site is a group of related content pages with a consistent look and feel, defined by a common layout, theme, and URL suffix.

Role required: content_admin or admin

Typically, all pages assigned to a site are given the attributes that are defined at the site level, such as the theme and layout. Individual pages can specify otherwise. At the least, a site needs the following fields defined: homepage, search results page, layout, URL suffix, and theme. All pages within a site can be accessed via the site URL suffix. The URL suffix is case-sensitive.

Associating a page with a site provides the following benefits:

- All pages within a site use the site default layout and theme, saving time.
- All pages within a site can be accessed using a uniform URL: customer.service-now.com/site-name/page.do.
- All pages within a site that have a search widget use the same search page.

Before creating a site for the first time, review the Content Management planning page to understand the concepts and principles of CMS.

1. Navigate to Content Management > Sites.
2. Click New.
3. Fill in the fields, as appropriate.

### Table 491: Site fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name for the site.</td>
</tr>
<tr>
<td>URL suffix</td>
<td>The URL suffix for the site. It is incorporated into the URL as follows:</td>
</tr>
<tr>
<td></td>
<td><a href="http://instance.service-now.com/url_suffix/page.do">http://instance.service-now.com/url_suffix/page.do</a></td>
</tr>
<tr>
<td>Home page</td>
<td>The page to display when the user does not specify a page name in the URL:</td>
</tr>
<tr>
<td></td>
<td><a href="http://instance.service-now.com/url_suffix/">http://instance.service-now.com/url_suffix/</a></td>
</tr>
<tr>
<td>Search page</td>
<td>The page that displays search results when a user searches from any page within the site.</td>
</tr>
<tr>
<td>Login page</td>
<td>The page to use for logging in to the site. If specified, users must log in to access the pages on the site. If left blank, no login is required to access the pages within the site.</td>
</tr>
<tr>
<td>Gauge target page</td>
<td>The page that displays gauge content. When the user clicks a gauge on the new site, the gauge target page opens showing the gauge content. The gauge target page replaces the CMS page in the current tab.</td>
</tr>
<tr>
<td>Title</td>
<td>A title for the site. The title can be the same as or different from the Name.</td>
</tr>
<tr>
<td>Description</td>
<td>A full description of the site.</td>
</tr>
<tr>
<td>Default layout</td>
<td>The layout for pages to use by default. Any page in the site that has a blank Layout field uses the layout set here.</td>
</tr>
<tr>
<td>Default theme</td>
<td>The theme for pages to use by default. Any page in the site that has a blank Theme field uses the theme set here.</td>
</tr>
<tr>
<td>Field</td>
<td>Input value</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Simple catalog display</td>
<td>Option to simplify catalog pages in the site by hiding the search bar, breadcrumbs, and the results per page choice list. Selecting this option prevents you from adding attachments from record producers to your CMS site.</td>
</tr>
<tr>
<td></td>
<td>Clear this check box to display the search bar, breadcrumbs, and the results per page choice list in catalog pages.</td>
</tr>
<tr>
<td>Use external cart</td>
<td>Option to omit the default cart when rendering catalog pages within a site that contains catalog pages. Provide a catalog cart block somewhere on the site to allow users to make catalog requests.</td>
</tr>
<tr>
<td></td>
<td>____________________________________________________________________________________________________________________________________________________________________________________________________________</td>
</tr>
</tbody>
</table>

The following is the site record for the ESS Portal:
You can add meta tags to a site. Meta tags are special tags that contain information about the site.

A DIV-based layout option is available. Many web page layouts use tables for a consistent look and feel. Tables are effective for numbers and statistics, but can be limiting for designing other types of information. DIV tags are flexible block-element tags. To use DIV tags for layouts, give the tag an ID and assign attributes using CSS.

Changing to CSS and DIV tags can:

- simplify code
- reduce the amount of code
- increase page load speed
- separate content from presentation
- help pages adapt to different device resolutions
- make pages easier for search engines to crawl
- make code more compliant with evolving web page design standards

A content theme named Administration Theme - Charcoal is included as an example of CSS-driven, DIV-based layouts.

To view the sample code in Administration Theme - Charcoal for a DIV-based layout:

1. Navigate to Content Management > Design > Layouts.
2. Click Admin 1 Column.

Copy a site

To create a site quickly, you can copy an existing site.

Role required: content_admin or admin

The site copy option creates a complete standalone copy of the site and all its resources. If you are copying a site to create a second site, it is best to use this option after the first site is complete, tested, and production ready. Copying a site includes pages, CSS, content blocks, and menus. Images are stored separately in the sys_attachments table and not included in the copy. There are various reasons why site copying is useful, such as site versioning, branding, or creating a backup.

To copy just a few pages without duplicating all the resources (CSS, blocks, menus), use the page copy option. Page copy duplicates the page but not the resources used in the page. For more information, see Copy a page.

1. Navigate to Content Management > Sites > [Site Name].
2. On the Site form, click Copy.
3. Type a name for the new site.

This name is a prefix for all the site elements duplicated. Do not use Portal or CMS in the site name.

A progress bar shows the copy process.

When the process is complete, the Site form shows information for the new site.

Content Management security

There are several methods for securing CMS sites and pages. Site security is set in the Login page field on the site record. You can control whether a page is public or private through the URL.
Every content page has its own URL that users can access outside of the platform. You can configure the URL to be public or private, based on the Login page and role definitions.

- If the content page has no defined Read role or there is no defined Login page, any internet user can navigate to the URL and view the content page.
- If there is a defined Read role, then anyone who goes to the URL is asked to log in before they can view the site.
- If there is a defined Login page on the site record, all pages in the site are private.

Site security is set in the Login page field on the site record (Content Management > Sites > [site name]). You can control whether a page is public or private through the URL.

Content Management URLs

The format for Content Management URLs is as follows.

```
<path to the instance> + /<site suffix> + /<page suffix> + .do
```

The `<site suffix>` is defined by the URL Suffix field on the site form.
The `<page suffix>` is defined by the URL Suffix field in the page form. The URL suffix is case-sensitive.

For example, the page **Austere - Site Entry** has a site URL Suffix of austere and a page URL Suffix of entry. The constructed URL looks like the following URL.

```
<instance name>.service-now.com/austere/entry.do
```

If the site URL Suffix field is left blank, the `<site suffix>` is cms, as shown in this example:

```
instance.service-now.com/cms/page.do
```

If the page URL Suffix is left blank, the name of the page is used as shown in this example:

```
instance.service-now.com/ess/Page Name.do
```

Special characters in the name of the page must be escaped.

Login pages instead of login rules

You set a login page on the site record to allow users to log in or out directly through the content site. Login rules were used in earlier versions to dictate what users saw after logging in, based on their roles or permissions. Login rules still work, but their use is deprecated.

Configure CMS sites for single sign-on (SSO)

To configure CMS to use SSO, make the view_content page private. Because public pages do not require login, and CMS sites are public by default, CMS pages do not use SSO. Use the following procedure to turn off public-facing content pages and enable SSO.

1. Navigate to Public Pages (sys_public.list).
2. For the view_content page, set Active to false.
Content pages

The core of the content management system is a page.

Pages are built from content blocks and organized into sites. Pages display as regular webpages that are rendered in HTML. Constructing a content page requires a basic knowledge of HTML. Content pages are formed by arranging content blocks in pre-defined layouts. They can be used to present login pages, search pages, or ServiceNow content.

Create a content page

The core of the content management system is a page.

Pages are built from content blocks and organized into sites. Pages display as regular webpages that are rendered in HTML. Constructing a content page requires a basic knowledge of HTML. Content pages are formed by arranging content blocks in pre-defined layouts. They can be used to present login pages, search pages, or ServiceNow content.

1. Navigate to **Content Management > Sites**.
2. Open the site that will contain the page.
3. In the **Pages** related list, click **New**.
4. Define the settings on the Page form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the page. As a best practice, prefix each page name with the name of the site followed by a dash and then the function of the page in the site. For example, <strong>ESS - Catalog Detail</strong>, <strong>ESS - Search Results</strong>, and <strong>ESS - Site Entry</strong> are all clear names for pages within the <strong>ESS</strong> site.</td>
</tr>
<tr>
<td>URL suffix</td>
<td>Enter the URL suffix for the page. The suffix is incorporated into the URL as follows: <a href="http://instance.service-now.com/site/url_suffix.do">http://instance.service-now.com/site/url_suffix.do</a></td>
</tr>
<tr>
<td>Parent page</td>
<td>Select the existing page that will be the parent of the current page. Parent pages keep sections within large sites sortable on the site list of pages. Parent pages are also used to dynamically create basic breadcrumb functionality. Use CSS to define menus that give the Parent page context within the user interface.</td>
</tr>
<tr>
<td>Layout</td>
<td>Select a layout to use for the page. Layouts define dropzones where content blocks can be added to the page. If this field is left blank, the page inherits the default layout of the site. If the site does not have a default layout, there is only a single dropzone for the entire page.</td>
</tr>
<tr>
<td>Field</td>
<td>Input value</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Content theme</td>
<td>Select a theme to use for the page. Themes bundle CSS style sheets that are applied to all content within the page.</td>
</tr>
<tr>
<td></td>
<td>If this field is left blank, the page inherits the default theme of the site. As a best practice, use the default theme unless the page requires a different set of CSS style sheets from the other pages in the site.</td>
</tr>
<tr>
<td>Frame buster</td>
<td>Select this option to remove any restrictions placed by frames that contain the page. Use this option to avoid frame-within-frame issues that can sometimes occur with improper linking.</td>
</tr>
<tr>
<td>Content site</td>
<td>Select the site associated with the content page. If you created this page from a related list, this information is automatically provided. The content site also determines part of the page URL, as follows:</td>
</tr>
<tr>
<td>Read roles</td>
<td>Click the lock icon to open a list for selecting roles that can view the page.</td>
</tr>
<tr>
<td>Model document</td>
<td>A document ID of a record to display by default.</td>
</tr>
<tr>
<td>Page status</td>
<td>Select a status for the page, such as Published or Retired.</td>
</tr>
<tr>
<td>Created by</td>
<td>Specify the user who created the page. If you are logged in with a role that has higher privileges than your user name and you enter your user name, the field defaults to the role. For example, if you are logged in as an Admin and you type your name, which has lower privileges, this field displays Admin.</td>
</tr>
<tr>
<td>Title</td>
<td>Enter a title for the page.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description of the page.</td>
</tr>
</tbody>
</table>

5. To add content blocks, click **Edit Page** under **Related Links** on the Page form.
6. Click **Add Content**.
7. Select a content block from the picker.
8. Select the dropzone where the content is to be added.

**Note:** You can create content blocks by adding one of the content blocks named "New [block type]" to the page.
Add content to a page

After defining the page settings, set the content of the page by adding content blocks the same way you add content to homepages.

2. Click Add Content.
3. Select a content block from the picker.
4. Select the dropzone where the content belongs.

**Note:** Create content blocks by adding one of the content blocks named "**New [block type]**" to the page.
Copy a page

Copying pages is an efficient way to avoid duplicating the same work and to create pages quickly from a guiding master template.

Role required: content_admin or admin

1. Navigate to Content Management > Sites and select the site.
2. Select the page to copy.
3. Click Copy.
4. Rename the page.

**Note:** Do not use service_catalog to rename a page. It is already a valid page in the system.

5. Edit the page fields.
6. Click Update.

Assign a page to a site

If you have created multiple sites, you can add pages from one site to another.

Role required: content_admin or admin

1. Navigate to Content Management > Sites.
2. Select a site.
3. In the Pages related list, click Edit.
4. Select other pages to include in the site.
5. Click Save.
6. Update the site.

Add a page to an application

Homepages and content pages are not added automatically to update sets and applications. They must be manually added.

Role required: content_admin or admin

1. Navigate to Content Management > Sites.
2. Right-click a content page record.

The page is added to the current application and to the current update set.

Configure a private UI page for CMS links

With single sign-on (SSO) in place, you can generate email links that take users directly to tickets and applications through the CMS interface.

Role required: content_admin or admin

You accomplish this procedure by creating a private UI page to redirect CMS links. Because the page is not public, it requires authentication and redirects to SSO appropriately.

1. Check that your CMS site is private by completing the following steps.
   a) Navigate to Content Management > Sites > [Your Site]
   b) Verify that the Login page field has a page listed.
2. Create a UI page by completing the following steps.
   a) Navigate to System UI > UI Pages.
   b) Click New.
   c) Enter a Name, for example, redirector.
   d) Enter the following Adobe Jelly code into the HTML field.

```xml
<?xml version="1.0" encoding="utf-8"?><j:jelly trim="true"
 xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null"
 xmlns:g2="null"><script type="script/javascript">
 window.location.href="${sysparm_uri}";</script></j:jelly>
```

The CMS supports email links with following format:

http://<path to instance>/<UI page>.do?uri=/<CMS suffix>/<record type>.do%26sys_id=<record ID>

For example:

http://<instance name>.service-now.com/redirector.do?sysparm_uri=/ess/incident.do%26sys_id=46e3e949a9fe19810069b824ba2c761a

**Note:** If you are using SAML 2.0 update 1 for SSO, use the UI page saml_redirector instead of creating a UI page. For example, an email link to a SAML 2.0 redirector page would use the URL format: http://<instance name>.service-now.com/saml_redirector.do?sysparm_uri=/ess/incident.do%26sys_id=46e3e949a9fe19810069b824ba2c761a.

### Content templates

Templates are content pages that are reused to provide consistent look and feel.

Templates are useful when creating CMS sites for the following reasons.

<table>
<thead>
<tr>
<th>Create pages rapidly without risk by copying pages</th>
<th>Using a template to create pages saves time because you do not have to keep repeating the same steps.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide a restore point for pages within the system</td>
<td>Have a working template available if something goes wrong in one of the blocks or in the theme CSS. It is often easier to start over from the template instead of trying to undo complex changes.</td>
</tr>
<tr>
<td>Provide a functional reference for editors on the site project</td>
<td>Having the template as a reference ensures that pages continue to conform to the style guide.</td>
</tr>
</tbody>
</table>

### Configure a master template

An easy way to create a master template is to copy the existing ESS sample site and customize it to suit your own needs. You can also configure a master template from scratch.

**Role required:** content_admin or admin

Follow these steps to create a single master page and generate all important components within the site.

1. Design a layout.
Regardless of the interface, a site can be distilled into a few simple layouts. For more information, see \textit{Layout concepts} on page 494.

2. Create a theme.
   The theme defines the structure of the layouts in CSS and the base styles, such as fonts and colors. For more information, see \textit{Content themes} on page 2106.

3. Build the common blocks.
   Pages are composed of content blocks. Most content blocks are reused on multiple pages. For the master template, build basic blocks such as a header, side navigation, and some basic content for the main content area of the page. More detailed content can be added later, but define content for reuse on many of the site pages here. For more information, see \textit{Content blocks} on page 2072.

4. Build a site entry page.
   Use the common content blocks you just created to design the first page that users see when they enter the site. For more information, see \textit{Create a content page} on page 2058.

5. Build a detail page.
   Design the detail pages to determine how pages such as knowledge articles, catalog items, and search results are displayed. You can build a detail page by copying the site entry page and adding additional content blocks. For more information, see \textit{Copy a page} on page 2064.

6. Assign the pages created to the new site.
   Create the site and apply the layout and theme to the site defaults. Then, navigate to the \textbf{All Pages} list and specify the master template site in the \textbf{Site} column for each of the master template pages. For more information, see \textit{Create a site} on page 2052.

\section*{Customize a copy of a page template}

One good source for templates is the base system sample site. It is easy to copy pages and restyle them into new page templates to meet different business requirements.

The Employee Self-Service portal is provided as a working example and design template in the following procedure.

1. Navigate to \textbf{Content Management} > \textbf{Sites} > \textbf{Employee Self-Service}.
2. Select a page.
   For example, to copy the ESS sample site homepage, select Portal.
3. Click \textbf{Copy}.
4. Rename the page.
5. Right-click in the header and select \textbf{Save}.
7. Edit the page by clicking the Edit icon \( 	ext{edit} \). For example, change the menus, alter the layout, or add a different logo.
8. Click \textbf{Update}.
9. Use the page in other sites you have created by \textit{assigning pages} to a site.

\section*{Page templates}

Every page that is part of the site needs a template. Sample page templates are listed here.
Critical page reference fields

When you build a new site, four page reference fields on the site record are critical. These pages are the first steps in understanding page templates and defining site defaults.

<table>
<thead>
<tr>
<th>Home page</th>
<th>landing page for the site. A home page is mandatory.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search page</td>
<td>page that displays search results when a user searches from any page within the site. A search page is mandatory for your site to have a Search Results block.</td>
</tr>
<tr>
<td>Login page</td>
<td>a standalone login page, which is useful to force authentication for the entire CMS site.</td>
</tr>
<tr>
<td>Gauge target page</td>
<td>page used to display the drill-through content from a gauge. When the user clicks a gauge, the gauge target page opens showing the drill-through content for that gauge.</td>
</tr>
</tbody>
</table>

Figure 565: Critical page reference fields
**Detail pages**

Depending on the data or tables you plan to use, you may need detail pages. These are pages defined by content types:

<table>
<thead>
<tr>
<th>Content Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge detail page</td>
<td>displays a full knowledge article detail (mandatory for a knowledge site).</td>
</tr>
<tr>
<td>Incident page</td>
<td>detail page for an incident record.</td>
</tr>
<tr>
<td>Catalog page</td>
<td>detail page for all items, content items, order guides, and record producers.</td>
</tr>
</tbody>
</table>

**Templates for creating sections**

There are two page templates you can use when creating sections:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent page</td>
<td>keeps sections within large sites organized and sortable on the site list of pages. Parent pages are also used to create breadcrumbs dynamically.</td>
</tr>
<tr>
<td>Detail page</td>
<td>differs from the parent page in that the content area displays a full article or detail instead of a selection of related content. Detail pages must have a Current Document block.</td>
</tr>
</tbody>
</table>

These pages are mandatory if you plan on accessing system data. For example, if you plan on showing the service catalog in your CMS, you need the following items.

- A service catalog content type that references the [sc_cat_item table].
- A detail page that provides the full view of the item.

**New templates**

If the base system sample site pages are not suitable, you can create a template from scratch. Create a page and then use it as a template. For details, see *Create a Content Page*.

When saving a new template, include the word “template” in the page name.

**Content types**

Content types provide site-specific control of how system data defined by templates is rendered.

In the site, one page displays a list of knowledge articles, and another page displays catalog items or incidents or a combination of the two. Different themes can be used for types of content and each theme can provide different user interaction. Content types define the pages that display content from a table. Each content type corresponds to a table.

Every type of document that the CMS displays has an associated content type. Changing the content type requires knowledge of Apache Jelly scripting. However, the common content types (such as service catalog or knowledge base) come in the base system. Content types can be associated with particular sites. This association allows different sites to use different detail pages for the same content type.

Content types define three features of associated documents.
• What does a link to one of these documents look like? For example, if a list of these documents is displayed on a page, how does each entry appear?
• What does a detailed view of one of these documents look like?
• What detail page is used to display the document? This decision is important and is often an area of confusion for new CMS users.

The document content type determines the page that a list of documents points to, the list itself does not determine the page. The content from a link is displayed in a detail content block on a page. The content type determines in which detail block on a page the document content is displayed.

For example, the list block Catalog Top 5 displays the top five items in the Service Catalog table [sc_cat_item]. Because the table is [sc_cat_item], the content type sc_cat_item controls how the Catalog Top 5 list is displayed. Clicking any item in the list displays the Service Catalog Detail page with the item displayed on it according to the detail template script. These content types are applied:

• in search results to link to the correct page.
• in the Current Document block to display the current record.
• in links on lists and other places that link to record types.

Several content types are available by default for tables such as Catalog Item [sc_cat_item], Gauge [sys_gauge], and Page [content_page].

Content type defaults that are set for CMS can be overridden for individual sites. Use the Content Types related list on the site record to customize content types.

Create a content type

Users with the content_admin role can create a content type.

Role required: content_admin or admin

1. Navigate to Content Management > Content Types.
2. Click New.
3. Complete the Content Type form.

Table 492: Content Type form

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Select the table whose content is rendered in Content Management.</td>
</tr>
<tr>
<td>Content site</td>
<td>Select the site that uses this content type.</td>
</tr>
<tr>
<td>Media type</td>
<td>Enter one of the following media types to use with this content type or leave the field blank to use the UI11 interface.</td>
</tr>
<tr>
<td></td>
<td>• doctype: UI15 desktop interface only</td>
</tr>
<tr>
<td></td>
<td>• m: Smartphone interface only (not for CMS use)</td>
</tr>
<tr>
<td></td>
<td>• tablet: Tablet interface only (not for CMS use)</td>
</tr>
<tr>
<td>Default detail page</td>
<td>Select which page is loaded after a link is selected.</td>
</tr>
</tbody>
</table>
### CMS gauge support

Gauges are a graphical way to display information from an instance.

A gauge could, for example, show a bar chart breaking down all open incidents by category. Gauges are fully supported within the CMS system. This means:

1. You can put a gauge on a CMS page
2. You can control what happens when a user clicks a cell within that gauge

**Put a gauge on a CMS page**

You have control over the gauges on a CMS page.

1. Bring up the CMS page in edit mode.
2. Click **Add Content**.
3. Select the gauge you want to add.
4. Place the gauge on the page in the desired location.

**Control what happens on a click**

Since a CMS system uses multiple pages, you have to tell the system which page to use to display drill through content.

In the typical (non-CMS) ServiceNow system, when you click a bar in a bar chart or a wedge on a pie chart, you drill through and your current screen is replaced with a list of records meeting the criteria in the chart. For example, if you have a gauge of **Incidents by Category** and you click the bar labeled **Hardware**, you drill through to a list of all incidents with a category=hardware.

Within the CMS system, there is a similar drill through mechanism at work, but you must manually specify which page to display.

There are two different models of the drill through:

- use an in-place target frame which directs all content there
- specify an alternate page (gauge target page) to use to display the drill-through content

**Using an In-Place Target**

An in-place target is a named iframe on the same CMS page as the gauge. When the gauge is clicked, the drill-through content appears within that iframe rather than changing out the CMS page. The gauge (and the rest of the page other than the target) remain in place and active.

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gauge page</td>
<td>Select which page is used as the drill-through target for any gauge. If you display a gauge on a CMS page, then clicking the links loads the page specified here.</td>
</tr>
<tr>
<td>Summary Template</td>
<td>Write an XML script that determines how the list is displayed in the list block, if the link is displayed in a list block.</td>
</tr>
<tr>
<td>Detail Template</td>
<td>Write an XML script that determines how to display the associated information after a user clicks the link.</td>
</tr>
</tbody>
</table>
Using a Gauge Target Page

A gauge target page is a separate CMS page that is used to display the drill-through content from a gauge. When a gauge is clicked, the current CMS page is replaced with the gauge target page and the drill-through content is rendered within that second page.

- **Gauge Target Page**
  
  To set a site default, select **Gauge Target Page**, on the site configuration page. The site default is used as the drill-through target for any gauges in the system that do not have a more specific gauge page specified in their content type.

- **Gauge Page**
  
  To set a content-type specific page, you can specify a **Gauge Page**, on a content type page. Any gauges for this type of content then use that page for any drill through.

  **Note:** The gauge target on a particular content type overrides the default, site level, gauge target.

- **Gauge Target block**
  
  A gauge target page is a normal CMS page, with one special requirement. Somewhere on that page, there must be a **Gauge Target block**. This is a market block telling the system where to output the drill-through data.

View content types

The following is an example using Content Types.

This example uses the out-of-box Employee Self-Service (ESS) site as an example.

Role required: content_admin or admin

Use any out-of-box instance of a CMS site.

2. Under Get Help, click **Issue Status**.

   The Issue Status, or incident_status CMS page contains two content blocks: Common Answer and Current Issues. These content blocks both use Content Types to render results.

3. From [https://<instance name>.service-now.com](https://<instance name>.service-now.com), navigate to **Blocks > Lists > Portal - Common Answers**.

   This block matches the Common Answers section of the Issue Status page. The Table field describes the Content Type associated with Common Answers.

4. Use the link to view existing Content Types.
5. Click the Content Type to view the XML that determines how the list and record are rendered in CMS.

   The Default Detail Page field indicates a CMS page that displays a record selected from a List of Content block.

6. Click the Information icon next to the Default Detail Page, to go to that page.
7. From the Default Detail Page, under Related Links, click **Edit Page**.

   The Detail Block contains a content type formatter. When editing or viewing the page, it is looking for a URL to be passed to determine the record to display, which is why it says "Detail record could not be located".

8. Click the pencil icon to edit and view the Detailed Content block.

   The Type drop down list has "Show the page's current document" selected, which indicates that the document_id must be passed to this block from the CMS pages, then it can display the record based on the content type listed in the document_id.
9. From the Common Answers block, on the Self-Service page, select a record.

The page opens with a URL similar to: https://<instance name>.service-now.com/ess/knowledge.do?
sysparm_document_key=kb_knowledge,02255450d731310013ab49547e61038e

The table `sysparm_document_key=kb_knowledge` and `sys_id 02255450d731310013ab49547e61038e` determine the record. The associated content type, `kb_knowledge`, renders the content on the Default Detail Page, Portal - Knowledge Detail / url_suffix=knowledge.

Content blocks

A block is a defined piece of content within the system that can be reused.

A content page is constructed by arranging customized blocks of content on a page. Existing content such as reports, gauges, and record lists are automatically available as content blocks and more can be created within the CMS.

After defining content blocks, use them on any content page by adding them to dropzones. For more information, see Add content to a page on page 2061.

Create content blocks after sites and pages have been designed.

Create a content block

To create a content block, you must define it in the appropriate form.

Access the form in one of these ways:

- Navigate to Content Management > Blocks > [Block type] > New.
  The form opens for creating the selected block type.
- If you navigate to Content Management > Blocks > All > New, a wizard guides you to the appropriate form.
- Edit a page and add a stub block of the desired type (*New [block type]). Then, click the link in the stub block.
  After you save the form, add the content block to any content page.

Create a detailed content block

A detailed content block displays the content of an existing document, such as a knowledge article or a service catalog request, as a block on a content page.

Role required: content_admin or admin

The block works with content types in the Default detail page field. For more information, see Create a content type on page 2069.

If you plan to use a script to find a document, configure the form to add the Script field if it is not displayed.

1. Navigate to Content Management > Configuration > Page Detail Settings.
2. Click New.
3. Complete the Detailed Content form.

Table 493: Detailed Content form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for the Detailed Content block.</td>
</tr>
</tbody>
</table>
Create a CMS block tag

A CMS block tag is used for advanced block creation and site flexibility.

Role required: content_admin or admin

It is constructed as `&lt;g:content_block&gt; {{Jelly_Tags|Jelly}}&gt;` and can be used in either of the following ways.

- Blocks: to display a block inside a block.
- Layouts: to display a block inside a layout.

The tag appears in the format, `&lt;g:content_block type="&lt;type&gt;" id="&lt;sys_id&gt;"/&gt;`.

An example of the block tag is included in the ESS Portal sample site.

1. Navigate to Content Management > Design > Frames.
2. Click `cms_admin_home_frame`.

© 2017 ServiceNow. All rights reserved. 2073
3. View the code.

```html
<style>
DIV.cms_administration_home {
  background: url(gray_${current_page.getURLSuffix()}.pngx) no-repeat right top;
}
</style>

<div class="cms_administration_home">
${body}
<br/>
<!-- Would you like to pivot off of parent page instead? try this snippet
<j:if test="${current_page.getParentPage().getURLSuffix()=='administration'}">
  <g:content_block type="content_block_menu" id="7afc342def002000914304167b2256ac"/>
</j:if>
  The defaults use the page URL suffix to define sub menus
  -->
  <j:if test="${current_page.getURLSuffix()=='administration'}">
    <g:content_block type="content_block_menu" id="7afc342def002000914304167b2256ac"/>
  </j:if>
  <j:if test="${current_page.getURLSuffix()=='community_inspired'}">
    <g:requires name="ess.portal.globals.jsdbx"/>
    <g:content_block type="content_block_menu" id="ccd4b8c7efb70000914304167b22566e"/>
  </j:if>
<br/>
</div>
```

4. To see what the page looks like, navigate to the ESS Portal administration page. For example, http://instance name.service-now.com/ess/administration.do.

Configure a header block

A header block is a visual element placed at the top of pages. It provides a place for branding and other important site-wide functionality.

Some of the options that are available in headers include global search, text size toggle, menus for navigational purposes, and the user name with logout link.

1. Navigate to Content Management > Blocks > Headers.
2. Click New.
3. Complete the Header form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the header block.</td>
</tr>
<tr>
<td>Field</td>
<td>Input value</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Background    | Select a background for the header logo and menus. Choices are:  
|               | • None - No background is rendered  
|               | • Gradient/Image - Specify an image or gradient image to serve as the background  
|               | • Colored Bars - Specify colors for three different sections of the header (top, middle, and bottom)                                            |
| Image         | If Background is Gradient/Image, use this field to upload an image for the background of the header block. For gradients, upload a gradient image. |
| Top bar color | If Background is Colored Bars, enter a CSS color or color name to use as the background for the top menu.                                     |
| Middle bar color | If Background is Colored Bars, enter a CSS color or color name to use as the background for the logo and text.                               |
| Bottom bar color | If Background is Colored Bars, enter a CSS color or color name to use as the background for the bottom menu.                                |
| Logo          | Select an image to serve as a logo. The image is also a link to the main page.                                                               |
| Active        | Select this check box to make the block available for use.                                                                                   |
| Text          | Enter text that displays next to the logo.                                                                                                   |
| Conditional   | Enter any scripted conditions to be applied. If selected, adds a Condition script field to the form.                                           |
| Category      | Select a category to provide organization for the header block. The category also determines the detail page in which header block links open.  
|               | (Detail pages often display information in different ways). Default options include:  
|               | • None  
|               | • General  
|               | • Knowledge Base  
<p>|               | • Service Catalog                                                                                                                             |
| Top Menu      | Select a navigational menu block to use as a menu above the logo.                                                                            |
| Bottom Menu   | Select a navigational menu block to use as a menu below the logo.                                                                          |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>Specify the height for the header.</td>
</tr>
<tr>
<td>Search</td>
<td>Select this check box to add a search element in the header.</td>
</tr>
<tr>
<td>Font sizer</td>
<td>Select this check box to include text sizing controls in the header.</td>
</tr>
<tr>
<td>Login</td>
<td>Select this check box to include a login link in the header. If the user is logged in, this element displays the user name and a logout link. If you specify a login page on the site record, remember to include a login link so users have a place to enter a username and password.</td>
</tr>
<tr>
<td>Chat Queue</td>
<td>Select the chat queue that users access by clicking the Help Desk Chat button in the header. Clear the field to remove the button. This field appears only if the Chat plugin is active.</td>
</tr>
</tbody>
</table>

Create a navigation menu block

Navigation menu blocks enable you to create a menu of links to different content pages.

Role required: content_admin or admin

The menu block defines the entire menu block and how it is displayed.

1. Navigate to **Content Management > Navigation Menus** and click **New**.
2. Click to select one of the CMS menu types displayed to open the Navigation Menu form.

**Note:** Older instances open the Navigation Menu form directly.
3. Fill out the Navigation Menu form to define the block.

**Table 495: Navigation Menu form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the block.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the type of navigation menu, which determines how the links are displayed.</td>
</tr>
<tr>
<td>Frame</td>
<td>Select a border style for the navigation menu block.</td>
</tr>
<tr>
<td>Conditional</td>
<td>Select this check box to enable the use of scripted conditions. If selected, a Condition script field is added to the form, along with the Logged On check box.</td>
</tr>
<tr>
<td>Logged on</td>
<td>Select this check box to display the navigation menu block only if the user is logged on. This field appears only if Conditional is selected.</td>
</tr>
</tbody>
</table>
### Field
<table>
<thead>
<tr>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition Enter a script to define the conditions for the navigation menu block. This field appears only if Conditional is selected.</td>
</tr>
<tr>
<td>Active Select this check box to make the block available for use.</td>
</tr>
<tr>
<td>Category Select a category to provide organization for the navigation menu block. The category also determines the detail page in which menu links open.</td>
</tr>
<tr>
<td>Align sections with tab Select this check box to align sections with tab.</td>
</tr>
</tbody>
</table>

**Navigation menus and content links**

The methods for choosing a link target (Open in: current window, iframe, or new window) and referencing the item linked (Redirect to: page reference, attachment reference, or URL reference) are similar, so learning how to link within any of these elements is useful.

- A content page reference helps you select the desired page.
  - For example, the value home.do links to the site homepage.
- An attachment reference allows you to reference a single file attached to the menu item or section record. If there is more than one attachment, only the first is referenced.
  - Link is generated automatically and displays a browser file save prompt.
- The specified URL reference allows you to link to a full URL string that may be in the ServiceNow system or part of another system.
  - For example, com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=66c313e7c0a8016b008ee1a8e3d97f5&sysparm_nameofstack=b654d15b6f921000914304167b025 links to the Ask a Question record producer.
  - If you need to link to a page in an existing intranet system, this attribute can be useful.

**Menu sections**

Menu sections define groups of links and how the links behave. For example, link behavior determines which page opens when the link is clicked and how it opens (such as in a new page or a new frame). Use the following procedure to view a sample menu section.

1. Navigate to **Content Management** > **Blocks** > **Navigation Menus**.
2. Click **Portal - Block Menu**.
3. In **Menu Sections**, click **Order Things**.
4. Click the reference icon next to **Detail Page**.
5. In **Related Links**, click **Edit Page**.

By default, the link goes to the Order Things page that has the Portal - Order Splash Menu (Vertical Block Menu) in the content area of the page.
In the majority of the default CMS menus, the menu section serves as the menu title followed by secondary text, a separation line, and whatever links you define as menu items. The two menus primarily illustrate CMS menus. Although the design is versatile and flows well between pages, there could be questions about linking and scalability to large catalogs. This may only be a good practice for a small catalog with limited items. Look at the request catalog list / grid view demo in the code example demos menu on the ESS Portal:

1. Open your instance.
2. Add /ess/manage.do to the address in your browser address bar. The address should look similar to: https://yourinstancename.service-now.com/ess/manage.do.
3. In the Code Example Demos section, click Request Catalog List / Grid View.
4. Browse the catalog items.

Menu items and content links

Menu items can be seen as featured links from each section. There are many choices in URL definitions and link options. Links within the ESS Portal all link to content pages within the site. In turn, each page has an iFrame or set of blocks that houses the corresponding data. This method is useful for rapid prototyping, but consider where you need to create dynamic detail pages. Looking across all the menu sections, this discussion explains some of the linking options available to records within the system.

Figure 566: Menu item

Before proceeding to the links themselves, it is important to mention content links in the CMS. Menu items are similar to content links except they are called by the list block, not as part of a navigation menu block. Menu items and content links function the same as far as how they are defined and the options available for linking to items within the ServiceNow system. Content links are meant to be called through a list block that calls the Content Link [content_link] table. Content links do not have the Logged in field or the Roles option that can be very useful for controlling the UI experience for various roles defined in the system.
Figure 567: Content link

**Linking to catalog items, reports, and business services**

![Business Services](image)

![Featured Services](image)

![Reporting](image)

**Figure 568: Integration points**

**Business Services** links to a content page (CMS page referenced: Business Service Portfolio, URL: business_service_category.do) that pulls the system service catalog homepage into a frame within the content area. Each link within this section leverages the browse by category page, where you pass in the name of the category to return results.
• Target page's iFrame URL: catalog_home.do?
sysparm_nameofstack=aabdae07ef221000914304167b22567d&sysparm_view=business&sysparm_clear_stack=yes
• Target page's frame name: gsft_main
  • Desktop Computing URL: category_browse.do?category=Desktop Computing
  • Business Applications URL: category_browse.do?category=Business Applications
  • Communications Services URL: category_browse.do?category=Communications Services
  • Infrastructure Services URL: category_browse.do?category=Infrastructure Services
  • Hosting Services URL: category_browse.do?category=Hosting Services

Featured Services links to to a content page which pulls a small subset of services into an iFrame.
• iFrame URL: com.glideapp.servicecatalog_category_view.do?
sysparm_parent=d67c446ec0a8016500335aa37eafbc1&sysparm_view=
• Frame name: gsft_main
  • Install Software URL: catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?
sysparm_id=10d69689c611227600ffe6a41c664824
  • Email Account URL: catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?
sysparm_id=d67a86b8ce080165009386c752cd4a09
  • Electronic Messaging URL: catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?
sysparm_id=533798810a0a0b600f1a03593e19058
  • VPN RSA Token URL: catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?
sysparm_id=d67b099ac0a80165019d0c276b772502
  • Shared Storage (SAN) URL: catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?
sysparm_id=cedd458a0a0b8300c3b1e32e7a3ac2

Reporting links to a content page that pulls the reports page into an iFrame. All links within this menu leverage homepages in the system, which creates an issue with the home.do URL. Notice in the links below that ../ is used to create a relative URL outside of the CMS site home.do definition. Without this, the site homepage would render within the iFrame.
• iFrame URL: report_home.do
• Frame name: gsft_main
  • Cost Management Overview URL: ../home.do?
sysparm_userid=0a81ae91c0a805c64c0942ab2e4b852b
  • Administration Overview URL: ../home.do?
sysparm_userid=0b7b11f6c611228901ff3cfb0b3cc8f
  • Portfolio Overview URL: catalog_home.do?sysparm_view=business
  • Service Availability URL: ../home.do?
sysparm_userid=0ee772000a0b0bad0c38eb7e68b93d0
  • Service Level Agreements (SLA) URL: ../home.do?
sysparm_userid=075e86a30a0006d4010a6851639498d1

View menu and list examples
Menu sections define groups of links and how the links behave.

Role required: content_admin or admin

For example, link behavior determines which page opens when the link is clicked and how it opens, such as in a new page or a new frame. Use the following procedure to view a sample menu section.

2. Click Portal - Block Menu.
3. In **Menu Sections** related list, click **Order Things**.
4. Click the reference icon next to **Detail Page** to open the page.
5. In **Related Links**, click **Edit Page**.

By default, the link goes to the Order Things page with the Portal - Order Splash Menu (Vertical Block Menu) in the page content area.

In most default CMS menus, the menu section has the menu title with secondary text, a separation line, and the links you define as menu items. Although the design is versatile and flows well between pages, there could be questions about linking and scalability to large catalogs. This practice may only be useful for a small catalog with limited items.

6. To view an example of dynamic content, follow these steps to look at the request catalog list/grid view on the ESS Portal.
   a) Add `/ess/manage.do` to your instance URL.
      For example, `https://<instance name>.service-now.com/ess/manage.do`.
   b) In the **Code Example Demos** section, click **Request Catalog List / Grid View**.
   c) Browse the catalog items.

*Menu style customization*

You can make menu style customizations in the style sheets your site uses.

Navigate to **Content Management > Design > Style Sheets** to customize style sheets.

**Supplementary page navigation menu example code**

In the example below, look at the CSS class selectors and rules. Also, note how the block containers (div.cms_menu_section_blocks) are defined based on the outer container (TD.layout_content_submenu_column). The outer container is actually a part of the site layout. This is clearer if you compare the styles to those used in the super menu further down in the example. The lesson is that there is no need to make a completely new menu system - just use CSS to change the look and feel of existing menus.

```html
/* *****************************************************************************************************/
SUB MENU VARIATIONS FOR HORIZONTAL MENUS - Section Blocks Menu
(cms_menu_section_blocks UI Macro)
*****************************************************************************************************/

TD.layout_content_submenu_column
  DIV.cms_menu_section_blocks{width:156px;height:auto;float:left;position:relative;border-style:solid;margin:0px0px0px0px;border:0pxsolid#e0e0e0;padding:0px;background:none;padding:24px12px0px12px;}

TD.layout_content_submenu_column
  DIV.sub_menu_section{width:156px;height:20px;float:left;border-style:solid;border:0pxsolid#e0e0e0;padding:0px;background:none;padding:0px12px0px12px;}

SPAN.cms_sub_menu_list_link, TD.submenu_cell A {color:#FFF;}

TD.layout_content_submenu_column IMG.menu_bullet{display:none;}

TD.cms_menu_section_blocks_title h2, TD.cms_menu_section_blocks_title h2 a,
  TD.cms_menu_tab_blocks_text h2 {margin:0;padding:0px;font-size:11px;text-transform:uppercase;color:#42C4DD;font-weight:normal;white-space:nowrap;}

TD.layout_content_submenu_column
  a.cms_menu_block_item{margin:0;padding:0px;font-size:11px;color:#FFF;}
```

© 2017 ServiceNow. All rights reserved. 2082
Super menu sections example code

This menu is essentially a simple float grid. The defaults are written first. Then, below the "SUPER MENU VARIATIONS..." comment, the defaults are overwritten by adding a containing div with a unique class.

In the code that renders the header for the base system, the bottom menu resides in a table cell with the class of "cms_header_bottom_menu" (note the TD.cms_header_bottom_menu CSS selector).

```html
/********************
Section Blocks Menu (cms_menu_section_blocks UI Macro)
/********************
```

```html
div.cms_menu_section_blocks{width:260px;height:260px;float:left;border-style:solid;margin:0px0px12px12px;border:1pxsolid#e0e0e0;padding:10px;background:url(blue/portal_horizontal_bkg.pngx)repeat-xcenterbottom;}
```

```html
p.cms_menu_separator{border-top:1pxdotted#ccc;margin-top:6px;margin-bottom:6px;}
```

```html
td.cms_menu_section_blocks_title h2, td.cms_menu_section_blocks_title h2 a, td.cms_menu_tab_blocks_text h2 {margin:0;padding:0px;font-size:larger;font-weight:normal;color:#444;}
a.cms_menu_block_item{margin:0;padding:0px;color:#999;font-size:inherit;}
```

```html
/********************
SUPER MENU VARIATIONS FOR HORIZONTAL MENUS - Section Blocks Menu (cms_menu_section_blocks UI Macro)
/********************
```

```html
TD.cms_menu_section_blocks_title h2, td.cms_menu_tab_blocks_text h2 {margin:0;padding:0px;font-size:larger;font-weight:normal;color:#444;}
a.cms_menu_block_item{margin:0;padding:0px;color:#999;font-size:inherit;}
```

```html
TD.cms_header_bottom_menu/*style the super menu drop down bar*/z-index:199;float:left;background:none;margin-left:44px;}
```

```html
TD.cms_menu_section_blocks{width:200px;float:left;border:0px;margin:0px0px12px0px;padding:0px;background:none;}
```

```html
TD.cms_menu_section_blocks_title h2, td.cms_menu_section_blocks_title h2 a, td.cms_menu_tab_blocks_text h2 {margin:0;padding:0px;font-size:larger;font-weight:normal;color:#444;}
a.cms_menu_block_item{margin:0;padding:0px;color:#999;font-size:inherit;}
```

```html
TD.cms_menu_section_blocks_title h2, td.cms_menu_tab_blocks_text h2 {margin:0;padding:0px;font-size:larger;font-weight:normal;color:#444;}
a.cms_menu_block_item{margin:0;padding:0px;color:#999;font-size:inherit;}
```

```html
TD.cms_header_bottom_menu/*style an item on the super menu drop down*/z-index:200;float:left;padding-left:12px;padding-right:12px;padding-bottom:8px;padding-top:4px;cursor:pointer;color:#000;border-left:1pxsolid#FFF;border-top:1pxsolid#FFF;}
```

```html
TD.cms_menu_section_blocks_title h2, td.cms_menu_tab_blocks_text h2 {margin:0;padding:0px;font-size:larger;font-weight:normal;color:#444;}
a.cms_menu_block_item{margin:0;padding:0px;color:#999;font-size:inherit;}
```

```html
TD.cms_menu_section_blocks_title h2, td.cms_menu_tab_blocks_text h2 {margin:0;padding:0px;font-size:larger;font-weight:normal;color:#444;}
a.cms_menu_block_item{margin:0;padding:0px;color:#999;font-size:inherit;}
```

```html
TD.cms_header_bottom_menu/*style a selected item on the super menu drop down*/z-index:200;float:left;padding-left:12px;padding-right:12px;padding-bottom:8px;padding-top:4px;cursor:pointer;color:
```

© 2017 ServiceNow. All rights reserved. 2083
Menu types

By changing the **Type** field on the navigation menu block, you can format the same menu in different ways.

Table 496: Menu types

<table>
<thead>
<tr>
<th>Type</th>
<th>Image</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop-Down Menu</td>
<td><img src="image" alt="Drop-Down Menu Image" /></td>
<td>The drop-down menu renders the menu sections as drop-down list. Use the mouse to hover over the menu name and view the menu items.</td>
</tr>
<tr>
<td>Type</td>
<td>Image</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Tab Menu for Headers</td>
<td><img src="image1.jpg" alt="Image" /></td>
<td>The tab menu renders the menu sections as tabs. Use the mouse to click the tab and view the menu items.</td>
</tr>
<tr>
<td>Horizontal Blocks</td>
<td><img src="image2.jpg" alt="Image" /></td>
<td>The horizontal blocks menu renders the menu sections as block headings with menu items as links within the blocks. The blocks are arranged horizontally.</td>
</tr>
<tr>
<td>Type</td>
<td>Image</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Tab Content Block</td>
<td><img src="image1.png" alt="Tab Content Block Image" /></td>
<td>The tab content block (horizontal) menu renders the menu sections as tabs with menu items as links within the block.</td>
</tr>
<tr>
<td>Vertical List</td>
<td><img src="image2.png" alt="Vertical List Image" /></td>
<td>The vertical list menu renders the menu sections as headings with menu items as links below them.</td>
</tr>
<tr>
<td>Type</td>
<td>Image</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Vertical Blocks</td>
<td><img src="image" alt="Image" /></td>
<td>The vertical blocks menu renders the menu sections as block headings with menu items as links within the block. The blocks are arranged vertically.</td>
</tr>
<tr>
<td>Super Menu</td>
<td><img src="image" alt="Image" /></td>
<td>The super menu is a hybrid between the drop-down menu and the tabbed system that allows the user to create a menu out of any number of menus.</td>
</tr>
</tbody>
</table>

**Configure menu sections**

Menu sections define groups of links displayed within the navigation menu block.

Role required: content_admin or admin

1. Navigate to **Content Management > Blocks > Navigation Menus** and select a block.
2. In the Menu Sections related list, click **New**.
3. Complete the Menu Section form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the menu section.</td>
</tr>
<tr>
<td>Content block menu</td>
<td>Select the navigation menu on which this menu section appears.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Redirect to</td>
<td>Select what appears when a user clicks the menu section name and icon.</td>
</tr>
<tr>
<td>URL</td>
<td>Click the lock icon to open the edit field, then enter the URL to display when a user clicks the menu section title and icon. This field is available only if Redirect to is set to The specified URL.</td>
</tr>
<tr>
<td>Detail page</td>
<td>Select the content page to open when a user clicks the menu section title or icon. This field is available only if Redirect To is set to A content page.</td>
</tr>
<tr>
<td>Left image DB</td>
<td>Select the image database where your site images are stored.</td>
</tr>
<tr>
<td>Right image DB</td>
<td></td>
</tr>
<tr>
<td>Left image</td>
<td>Select icons to appear on the left and on the right of the name.</td>
</tr>
<tr>
<td>Right image</td>
<td></td>
</tr>
<tr>
<td>Second level text</td>
<td>Enter a description to appear beside the menu section title.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the menu section available for use.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to indicate where this section appears on the menu relative to other menu sections.</td>
</tr>
<tr>
<td>Logged on</td>
<td>Select this check box to display the menu section only if the user is logged on.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock to open a list, then select the roles that can access this menu section if you restrict access by role.</td>
</tr>
<tr>
<td>Category</td>
<td>Select the category in which the menu section belongs.</td>
</tr>
<tr>
<td>Open In</td>
<td>Select the behavior of clicked links.</td>
</tr>
<tr>
<td>iFrame</td>
<td>Enter the name of the iFrame where the link opens when a user clicks the menu section title and icon. Make sure that there is an iFrame on the page where the link opens. This field is available only if Open In is set to Named iFrame.</td>
</tr>
<tr>
<td>Header</td>
<td>Enter header information for the menu section.</td>
</tr>
<tr>
<td>Footer</td>
<td>Enter footer information for the menu section.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.
**Menu sections**
Define each section of the menu by navigating to **Content Management > Navigation Menu**. On the Menu Sections related list, click **New**, then fill out the Menu Section form.

**Table 498: Menu Section form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the menu section.</td>
</tr>
<tr>
<td>Content block menu</td>
<td>Select the navigation menu on which this menu section appears.</td>
</tr>
<tr>
<td>Redirect to</td>
<td>Select what appears when a user clicks the menu section name and icon.</td>
</tr>
<tr>
<td>URL</td>
<td>Click the lock icon to open the edit field, then enter the URL to display when a user clicks the menu section title and icon. This field is available only if Redirect to is set to The specified URL.</td>
</tr>
<tr>
<td>Detail page</td>
<td>Select the content page to open when a user clicks the menu section title or icon. This field is available only if Redirect To is set to A content page.</td>
</tr>
<tr>
<td>Left image</td>
<td>Select an icon to appear on the left of the name.</td>
</tr>
<tr>
<td>Right image</td>
<td>Select an icon to appear on the right of the name.</td>
</tr>
<tr>
<td>Second level text</td>
<td>A description to appear next to the menu section title.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the menu section available for use.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to indicate where this section appears on the menu relative to other menu sections.</td>
</tr>
<tr>
<td>Logged on</td>
<td>Select this check box to display the menu section only if the user is logged on.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock to open a list, then select the roles that can access this menu section if you want to restrict access by role.</td>
</tr>
<tr>
<td>Category</td>
<td>Select the category in which the menu section belongs.</td>
</tr>
<tr>
<td>Open In</td>
<td>Select the behavior of clicked links.</td>
</tr>
<tr>
<td>iFrame</td>
<td>Enter the name of the iFrame where the link should open when a user clicks the menu section title and icon. Make sure that there is an iFrame on the page where the link opens. This field is available only if Open In is set to Named iFrame.</td>
</tr>
</tbody>
</table>
Configure menu items
Menu items are the links that appear within each menu section.

Role required: content_admin or admin

Not all navigation menu sections require menu items. Use menu items to link users to other pages or additional information. These steps use the ESS sample portal site as an example.

1. Navigate to **Content Management > Blocks > Navigation Menus** and select a navigation menu. For example, select the **Documentation Menu**.
2. From the **Menu Sections** related list, select a menu section. For example, select **Getting Started**.
3. From the **Menu Items** related list, click **New**.
4. Complete the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for the menu item.</td>
</tr>
<tr>
<td>Menu section</td>
<td>Select the menu section in which this item appears.</td>
</tr>
<tr>
<td>Redirect to</td>
<td>Select what appears when the menu item name and icon are clicked.</td>
</tr>
<tr>
<td>URL</td>
<td>Click the lock icon to open the edit field, then enter the URL to open when the menu item title and icon are clicked. This field is available only if <strong>Redirect To</strong> is set to <strong>The specified URL</strong>.</td>
</tr>
<tr>
<td>Detail page</td>
<td>Select the content page to open when the name or icon is clicked. This field is available only if <strong>Redirect To</strong> is set to <strong>A content page</strong>.</td>
</tr>
<tr>
<td>Image</td>
<td>Select an icon to appear with the name.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the menu item available for use.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to indicate where this item appears on the menu relative to other menu items.</td>
</tr>
<tr>
<td>Logged on</td>
<td>Select this check box to display the menu item only if the user is logged on.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock to open a list, then select the roles that can access this menu item to restrict access by role.</td>
</tr>
<tr>
<td>Category</td>
<td>Select the category in which the menu item belongs.</td>
</tr>
<tr>
<td>Open in</td>
<td>Select the behavior of clicked links.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>iFrame</td>
<td>Type the name of the iFrame where the link opens when a user clicks the menu item name and icon. Make sure that there is an iFrame on the page. This field is available only if Open In is set to Named iFrame.</td>
</tr>
</tbody>
</table>

5. Click **Submit**.

**Menu items**
Add menu items from the Menu Section form.

In the Menu items related list, click **New** and fill out the Menu Item form.

### Table 500: Menu Item form

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the menu item.</td>
</tr>
<tr>
<td>Menu section</td>
<td>Select the menu section in which this item will appear.</td>
</tr>
<tr>
<td>Redirect to</td>
<td>Select what appears when a user clicks the menu item name and icon.</td>
</tr>
<tr>
<td>URL</td>
<td>Click the lock icon to open the edit field, then enter the URL to display when a user clicks the menu item title and icon. This field is available only if Redirect to is set to The specified URL.</td>
</tr>
<tr>
<td>Detail page</td>
<td>Select the content page to open when a user clicks the name or icon.</td>
</tr>
<tr>
<td>Image</td>
<td>Select an icon to appear with the name.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the menu item available for use.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number to indicate where this item appears on the menu relative to other menu items.</td>
</tr>
<tr>
<td>Logged on</td>
<td>Select this check box to display the menu item only if the user is logged on.</td>
</tr>
<tr>
<td>Roles</td>
<td>Click the lock to open a list, then select the roles that can access this menu item if you want to restrict access by role.</td>
</tr>
<tr>
<td>Category</td>
<td>Select the category in which the menu item belongs.</td>
</tr>
<tr>
<td>Open in</td>
<td>Select the behavior of clicked links.</td>
</tr>
</tbody>
</table>
Configure dynamic blocks

Use dynamic blocks to use scripting or to pull information from the system. Dynamic blocks are where most of your content resides.

Role required: content_admin or admin

A good use of dynamic blocks is job postings. Store the postings in knowledge articles and display the postings with a dynamic block.

Several dynamic blocks are predefined, including the following items.

- **New Content**: Each of the new content blocks allows for creating blocks while editing content pages.
- **Clean Login**: The default login page requesting user name and password, which includes a "Remember Me" check box. After the user logs in, this block triggers login rules.
- **Login**: An area that allows a logged out user to log in and a logged in user to log out. This block is especially useful on publicly available content pages.
- **Search**: The global text search field. Currently, global text search is the only form of search that can be included in a content page.
- **Search Results**: An area for displaying global text search results.

For information about Apache Jelly, see *Content Management and the Apache Jelly engine* on page 2100.

1. Navigate to **Content Management > Dynamic**.
2. Click **New**.
3. Complete the Dynamic Content form.

### Table 501: Dynamic Content form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for the dynamic content block.</td>
</tr>
<tr>
<td>Category</td>
<td>Select a category to provide organization for the dynamic block. The category also determines the detail page in which dynamic block links open.</td>
</tr>
<tr>
<td>Frame</td>
<td>Select a border styling for the dynamic block. For more information, see <em>Format a frame</em> on page 2108.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the block available for use.</td>
</tr>
<tr>
<td>Conditional</td>
<td>Enter any scripted conditions to be applied. If selected, adds a Condition script field to the form.</td>
</tr>
</tbody>
</table>
### Catalog cart block

The catalog cart block is a dynamic block provided in the base system.

This block provides the same cart available within the service catalog. If the site definition has the Use external cart option selected, this block needs to be included in the site so users can interact with their catalog cart.

Catalog cart block is available when the glide.sc.use_cart_layouts property is set to false.

### Customize a list block

List blocks are content blocks that dynamically generate a list of links to records within the instance. When a user clicks a link in a list block, the associated information is displayed in a detail page determined by its content type.

Role required: content_admin or admin

Make the list using a simple query on any table or by scripting a more advanced query. Lists are powerful and flexible. Here are some places that allow you to have complete control over the list display.

#### Table 502: List control

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frames</td>
<td>Frames provide a method to create decorative containers for content blocks and other elements within the site. When frames are used with the Type field, the designer has complete control over list placement.</td>
</tr>
<tr>
<td>List Type</td>
<td>The List Definitions module defines the type of lists available for content pages. On the List Block form, select a list definition in the Type field.</td>
</tr>
<tr>
<td>List Filtering</td>
<td>Provides every field in the referenced table for more granular results.</td>
</tr>
<tr>
<td>Max Entries</td>
<td>Limits the results from a table to fit the design of the block.</td>
</tr>
<tr>
<td>Order and Order Direction</td>
<td>Allows sorting by any field in the referenced table, in either ascending or descending order.</td>
</tr>
</tbody>
</table>

You can also use list blocks to create a list of links to information outside your instance. Create the external links as records on the Content Link [content_link] table, and then follow the steps in this procedure.

1. To define a list block, navigate to **Content Management > Lists** and click **New**.
2. Complete the form.
# Table 503: List block fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the list content block. This name is used to identify the record in ServiceNow and is not displayed on the content page. Use the Title field for that purpose.</td>
</tr>
<tr>
<td>Category</td>
<td>Select a category to provide organization for the list content block. Default options include:</td>
</tr>
<tr>
<td></td>
<td>• None</td>
</tr>
<tr>
<td></td>
<td>• General</td>
</tr>
<tr>
<td></td>
<td>• Knowledge Base</td>
</tr>
<tr>
<td></td>
<td>• Service Catalog</td>
</tr>
<tr>
<td>Type</td>
<td>Select a list definition UI Macro to format the list of links. For more information on list definitions, see <em>List Definitions</em>.</td>
</tr>
<tr>
<td>Frame</td>
<td>Select a border style for the list block. For more information, see <em>Creating a New Frame</em>.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to enable generating a list from a script, rather than a simple filtered query on a particular table.</td>
</tr>
<tr>
<td>Query</td>
<td>Filters the results using a condition builder.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the block available for use.</td>
</tr>
<tr>
<td>Title</td>
<td>Enter the name to display at the top of the list block when it appears in a content page.</td>
</tr>
<tr>
<td>Maximum entries</td>
<td>Set the maximum number of entries to be displayed in this list block.</td>
</tr>
<tr>
<td>Table</td>
<td>Select a table to query for the list items. The table determines which detail page displays a user clicks a link in the list block. For more information, see <em>Configure the Content Type</em>.</td>
</tr>
</tbody>
</table>

**Note:** The list shows only tables and database views that are in the same *scope* as the list block.
3. Click **Submit**.

**Configuring content type in a list block**
Content types define how a list is displayed in a list block. Each content type corresponds to a table.
Select a table in the list block Table field to make that list block use the corresponding content type.
For instructions on how to define the content type, see **Create a content type** on page 2069.

**Configure list definitions**
List definitions, similar to content management frames, are decorative containers that control the look and feel of lists.
Role required: content_admin or admin
Specifically, list definitions are UI macros that use Adobe Jelly script to define how a list is rendered inside a list block. Site design often requires multiple list styles within the layout. Lists are often the primary form of navigation within a site, so it is important to have control over their formatting.
Configuring list definitions requires a knowledge of Adobe Jelly.

1. Navigate to **Content Management > Configuration > List Definitions.**
2. Click **New.**
3. Complete the List Definition form.

**Create a static HTML block**
Use static blocks for text that does not change. For example, use a static block for a site footer with only the company or organization name. A static HTML block allows any HTML code to be run within a page.
Role required: content_admin or admin
The HTML editor in the static block has limited functionality for designers who are familiar with editing HTML, Jelly, or Javascript. The HTML editor can also add tags or formats that advanced coders find unnecessary. For more advanced options, use **dynamic blocks**.

1. To create a new static HTML block, navigate to **Content Management > Static HTML.**
2. Click **New.**
3. Complete the Static Content form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the static HTML content block.</td>
</tr>
<tr>
<td>Field</td>
<td>Input value</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Category       | Select a category to provide organization for the static HTML block. The category also determines the detail page in which static HTML block links open. (Detail pages often display information in different ways). Default options include:  
  - None  
  - General  
  - Knowledge Base  
  - Service Catalog                                                                                                                                 |
| Frame          | Select a border styling for the static HTML block. For more information, see Creating a New Frame.                                                                                                          |
| Active         | Select this check box to make the block available for use.                                                                                                                                                  |
| Conditional    | Enter any scripted conditions to be applied. If selected, adds a Condition script field to the form.                                                                                            |
| Static Content | Enter HTML code that determines the behavior of the static HTML block.                                                                                                                                       |

**Format an image as a static HTML block**

An easy way to add an image to a CMS page is to use a static HTML block. After the block is created, you can reuse it throughout the site.

1. Navigate to Content Management > Design > Images.
2. Click New.
3. Select a Category to help organize the images.
4. Type the file name of the image, including the extension (such as .png).
5. Upload the file by selecting Click to add and browsing for the image.
6. Click OK.
7. Click Update.
9. Paste the following code into the HTML block, substituting the image name as uploaded in the previous step.
   The "x" at the end of the filename is required for image caching.

   ```html
   <img src="<image_name>.gifx"/>
   ```

Now the image is a static HTML block and you can add it to any content page. Use standard HTML code to alter the image in the content block.

**Configure Flash movie blocks**

Use a Flash movie block to embed any Flash movie (.swf file) as an attachment or by referencing a URL in a content page.
Role required: content_admin or admin

To add streaming video or Flash video (.flv), for example, to a knowledge article, see *Adding media to HTML fields* on page 806.

1. To create a Flash movie block, navigate to **Content Management > Flash Movies**.
2. Click **New**.
3. Complete the Flash Movie form.

**Table 505: Flash movie form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for the Flash movie block.</td>
</tr>
<tr>
<td>Source</td>
<td>Determines where the Flash movie is found. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• Attachment - If this choice is selected, <em>upload</em> the Flash movie to this record.</td>
</tr>
<tr>
<td></td>
<td>• Link to External Object - If this choice is selected, a URL field is displayed. Specify the Flash movie URL and ensure that the Flash movie is publicly accessible.</td>
</tr>
<tr>
<td>Height</td>
<td>Height, in pixels, for the Flash movie.</td>
</tr>
<tr>
<td>Width</td>
<td>Width, in pixels, for the Flash movie.</td>
</tr>
</tbody>
</table>

Configure content links

Content links are the predecessors to navigation menus. You can use content links to create navigational links to information outside of your instance.

Role required: content_admin or admin

After the content link is configured, it must be defined in a list block. List blocks use content links to reference information outside of your CMS.

1. Navigate to **Content Management > Specialty Content > Content Links**.
2. Click **New**.
3. Complete the Content Link form.

**Table 506: Content Link form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for the content link</td>
</tr>
<tr>
<td>Redirect to</td>
<td>Select the location of the external link</td>
</tr>
<tr>
<td>Category</td>
<td>Select a category to provide organization for the list content block.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
URL | Click the lock icon to open the edit field, then enter the site URL to open when the link is clicked. This field is available only if Redirect to is set to The specified URL.
Open in | Select the behavior of the clicked link.
Detail page | Select the page where the link opens when it is clicked. This field is available only if Redirect to is set to A content page.
iFrame | Enter the name of the iFrame where the link opens when it is clicked. Make sure that there is an iFrame on the page where the link opens. This field is available only if Open In is set to Named iFrame.

4. Click Submit.
5. From Content Management > Blocks > Lists, create a list block that references the content links by selecting Content Link in the table field.
6. Create a query to determine which external links display on the page. For example, the query Category is Search displays any external links that have a category defined as Search.
7. Click Update.

Define an iFrame

Define an iFrame with forms created in the system and link to the page in which the iFrame resides.

Role required: content_admin or admin

Configure an iFrame with forms you create in the instance and link to the page in which the iFrame resides.

1. Navigate to Content Management > Specialty Content > iFrames.
2. Click New.
3. Complete the iFrame form fields.

Table 507: iFrame form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for the iFrame block.</td>
</tr>
<tr>
<td>Frame name</td>
<td>Type a name for the frame on the page. Make sure that there is an iFrame on the page where the link opens. This name allows links within the iFrame to open within the iFrame.</td>
</tr>
</tbody>
</table>
Field | Input value
--- | ---
**URL** | Enter the URL to display in the iFrame. If you use the iFrame to display ServiceNow content, start with the page name and do not include the base instance part of the URL. For example, to show the list of requested items, the URL is: `sc_req_item_list.do`
Queries can be applied to the URL. For instance, to display a list of open requested items, the URL is: `sc_req_item_list.do?sysparm_query=active=true`
For more information, see [URL schema](#) on page 53.

**Sizing** | Select an option for iFrame block size.

*Note:* The Expand to fit content choice only works with ServiceNow content. If Fixed Size is selected, height and width fields are displayed for you to enter the size in pixels.

---

**Note:** Some browsers suppress iFrames because they use an X-Frame-Options header value of SAMEORIGIN. The X-Frame-Options header was introduced in Internet Explorer 8 RC1, to help detect and prevent frame-based redressing. The SAMEORIGIN value causes the browser to render a blank page instead of the target page of the `<frame>` or `<iframe>` when the frame target is not on the same origin as the page itself. Support for this header has been implemented in Safari 4.0, Chrome 4.1.249.1042, and Firefox 3.6.9 and above.

**Integrate Live Feed with CMS**

You can provide access to Live Feed from pages built in the Content Management System (CMS). For example, allow an end user to access your company feed via the ESS portal.

The ESS Portal template includes the Portal - Live page and Live Feed dynamic block (requires the Live Feed plugin). To provide access to Live Feed from CMS pages, *add the Live Feed dynamic block* to a CMS page or include the Portal - Live page in a site.

1. Navigate to **Content Management > Specialty Content > iFrames**.
2. Click **New**.
3. Complete the iFrame block form with the following values.

**Table 508: iFrame values**

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name: Live Frame</td>
</tr>
<tr>
<td>Frame name</td>
<td>Type a frame name: live_frame</td>
</tr>
</tbody>
</table>
Field | Input value
--- | ---
URL | Enter `https://instance name/live_feed.do?sysparm_doctype=true`. Replace `instance name` with your instance URL, for example, `<myinstance>.service-now.com`.

Sizing | Select **Fixed Size** and enter height and width pixel dimensions according to the page on which you plan to display the feed. For example, enter a width of 1024 and height of 768.

4. Click **Submit**.
5. Complete the steps in **Add the block to a page**.

*iFrame methods*

The following examples show how system records are pulled into an iFrame that is placed on a content page.

For system lists or forms, use the frame name **gsft_main** so that links work properly.

- **Order Hardware** is an example of linking to a catalog category.
  - **URL**: `com.glideapp.servicecatalog_category_view.do?sysparm_parent=d258b953c611227a0146101fb1be7c31&sysparm_view=`
  - Frame name: **gsft_main**

- **My Approvals List** is an example of linking to a list with a view filter and a JavaScript that reference the authenticated user.
  - **URL**: `sysapproval_approver_list.do?sysparm_query=approver=javascript:getMyApprovals()&sysparm_view=ess`
  - Frame name: **gsft_main**

- **Service Catalog Home Page (system)** references the system catalog page within the Service Catalog application. If you are satisfied with the way the catalog looks, this method is an easy way to bring the page into a CMS design.
  - **URL**: `catalog_home.do?sysparm_view=catalog_default`
  - Frame name: **gsft_main**

- **Problem Management Overview** references a homepage. `../` makes the URL string relative to system homepages. Without it, the URL string resolves to the default CMS homepage reference in the site.
  - **URL**: `../home.do?sysparm_view=problem_overview`
  - Frame name: **gsft_main**

**Content Management and the Apache Jelly engine**

Apache Jelly is a Java-based and XML-based scripting and processing engine for turning XML into executable code.

The Apache Jelly engine closely resembles XML and is typically comfortable for developers familiar with JavaScript, XML, XHTML, or HTML. In the ServiceNow instance, the Apache Jelly engine renders items such as forms, lists, and UI Pages. Apache Jelly code renders well within a dynamic content block, but can
Geneva
ServiceNow
ServiceNow Platform

have issues when used in static blocks. You can use Jelly tags, calls, and statements, but HTML acts just like XHTML.

```xml
<?xml version= "1.0" encoding= "utf-8" ?>
<j:jelly trim = "false" xmlns:j = "jelly:core" xmlns:g = "glide" xmlns:j2 = "null" xmlns:g2 = "null">
    <j:if test = "${current_page.getName()==='Solutions'}">
        <h1 class = "page_name" > <b> <a href = "solutions.do?" title = "${gs.getMessage('Solutions')}">${gs.getMessage('Solutions')}</a> </b> </h1>
        <p class = "page_description" > 
            ${current_page.getDescription()}
        </p>
    </j:if>
    <j:if test = "${current_page.getName()==='IT 3.0'}">
        <h1 class = "page_name" > <b> <a href = "solutions.do?" title = "${gs.getMessage('Solutions')}">${gs.getMessage('Solutions')}</a> </b> | 
            ${current_page.getName()})</h1>
        <p class = "page_description" > 
            ${current_page.getDescription()}
        </p>
    </j:if>
</j:jelly>
```

Ensure that all tags are closed. If the tag is not a naturally closing tag, then place a forward slash before the end bracket. For example, a <BR /> or an <IMG src="cms.png" />.

If you are unfamiliar with Jelly, review these topics:

- **Jelly tags** on page 3931
- **Extensions to Jelly syntax** on page 3937

Include the following tag with all Apache Jelly scripts.

```xml
<j:jelly trim= "false" xmlns:j= "jelly:core" xmlns:g= "glide" xmlns:j2= "null" xmlns:g2= "null">
```

The tag looks complex, but keep the following information in mind.

- Apache Jelly script uses multiple namespaces.
- There are two types of prefixes in tags: j and g. The j prefix is used for tags that are natively part of Apache Jelly. The g prefix is used for tags that the ServiceNow platform created and is using for platform purposes.

The j2 and g2 prefixes are just like j and g, except that they are processed in a second phase. The Apache Jelly script parser runs through each j and g tag respectively. For example:

```xml
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:g2="null">
    <j:set var="jvar_phase1" value="Hello" />
    <j2:set var="jvar_phase2" value="World" />
    ${jvar_phase1} ${jvar_phase2}
</j:jelly>
```

In phase 1, the parser runs through all the j and g tags. It then caches the result. Before it runs the second phase, it takes the j and g namespaces and moves the namespaces to the second phase. It looks something like the following code.

```xml
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="jelly:core" xmlns:g2="glide">
```
Hello $[jvar_phase2]
</j:jelly>

For subsequent calls of this script, only phase 2 is parsed.

**Note:** If you plan to use phase 2 Jelly tags (g2 and j2) on the Content Type [content_type] or Dynamic Content [content_block_programmatic] tables, select the **Two phase** option on the content form.

Another example is to create a report of all open incidents assigned to each group. For this purpose, you could use a report and save time, but it is a good example for learning Jelly. Start with the Jelly tag:

```jelly
<j:jelly trim= "false" xmlns:j= "jelly:core" xmlns:g= "glide" xmlns:j2= "null" xmlns:g2= "null">
</j:jelly>
```

First, you need a list of open incidents. Use a g2:evaluate tag. The evaluate tag runs the script. Anything inside the tag is parsed like a business rule, so, for example, you can call global business rules, script includes, and glide records.

```jelly
<j:jelly trim= "false" xmlns:j= "jelly:core" xmlns:g= "glide" xmlns:j2= "null" xmlns:g2= "null">
<g:evaluate var="jvar_groups" object="true">
 var gr = new GlideRecord("sys_user_group");
 gr.orderBy('name');
 gr.query();
 gr;
</g:evaluate>
</j:jelly>
```

This script is in phase 1 because frequent changes to incident assignment groups are not expected. Also notice the var attribute on the evaluate tag. This attribute specifies what variable is set from this block. At the end of the script, there is a gr on a line by itself. That last line is what sets the variable.

You can omit the jvar_groups variable, but then all the variables in the evaluate tag become Apache Jelly variables. The object=true specifies that the variable is not a primitive data type. If object=true is omitted, the script would break because jvar_groups would only be able to hold items like integers and strings.

After the evaluate tag, loop through these groups and find the incidents for each one.

```jelly
<j:jelly trim= "false" xmlns:j= "jelly:core" xmlns:g= "glide" xmlns:j2= "null" xmlns:g2= "null">
<g:evaluate var="jvar_groups" object="true">
 var gr = new GlideRecord("sys_user_group");
 gr.orderBy('name');
 gr.query();
 gr;
</g:evaluate>
<table>
<tr>
 <th>Name</th>
 <th>Incidents</th>
</tr>
<j:while test="${jvar_groups.next()}">
 <tr>
  <td>${HTML:jvar_groups.getValue('name')}</td>
  <td></td>
 </tr>
</j:while>
</table>
```
You can include normal XML in the Apache Jelly script at any time. Since there is no namespace, the Apache Jelly script does not try to parse the XML tags. Notice the \texttt{j:while} loop. It is a normal while loop and can iterate through a GlideRecord object. Also notice that you output a value with \texttt{${\text{HTML: jvar_groups.getValue('name')}}}. Here are the important elements:

- The outer brackets, \texttt{$\{$}, specify the output of the variable and the phase in which the variable is output: \texttt{$\{}$ means first phase, \texttt{$[]$} means second phase.
- HTML before the expression is for escaping the output. The expression \texttt{jvar_groups.getValue('name')} is being escaped for HTML. For other types of escaping, there are \texttt{JS} (Javascript), \texttt{NS} (No Script), and some other options.

To select only one record and not iterate through many records, the code looks like the following example:

```xml
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:g2="null">
  <g:evaluate var="jvar_groups" object="true">
    var gr = new GlideRecord("sys_user_group");
    gr.orderBy('name');
    gr.query();
    gr;
  </g:evaluate>
  <j:if test="${jvar_groups.next()}">
    We found ${\text{HTML: jvar_groups.getValue('name')}}
  </j:if>
</j:jelly>
```

### Content Management and Jelly code examples

#### Code examples

**Header Example Code**

This dynamic content block must be active, and the Two Phase option selected. The \texttt{g:requires} tag is including the UI script defined in the system whose name is "servicenow.website.globals." The file extension in the call is .jsdbx and is used only in the call to the UI script, not in the name of the script in the system. For JSDBX, the file being called is a JavaScript (.js) defined within the database (db) that is cached (x).

```xml
<?xml version= "1.0" encoding= "utf-8" ?><j:jelly trim = "false" xmlns:j = "jelly:core" xmlns:g = "glide" xmlns:j2 = "null" xmlns:g2 = "null">
  <g:requires name = "servicenow.website.globals.jsdbx" />
</j:jelly>
```

**Page Title and Description Example Code**

This dynamic content block must be active. There are two actions within this code snippet. First is a forward-looking string container that allows site translation, the \texttt{{$gs.getMessage('Your Text')}} string call). The second action pulls in the page title and description, \texttt{{$current_page.getName()}} and \texttt{{$current_page.getDescription()}}.

```xml
<?xml version= "1.0" encoding= "utf-8" ?><j:jelly trim = "false" xmlns:j = "jelly:core" xmlns:g = "glide" xmlns:j2 = "null" xmlns:g2 = "null">
  <j:if test = "${current_page.getName()='Solutions'}" >
    <h1 class = "page_name" > <b > <a href = "solutions.do?" title = "${gs.getMessage('Solutions')}">${gs.getMessage('Solutions')}</a> </b> </h1>
    <p class = "page_description" >
```
List Block Pulling From Knowledge Articles Example Code

This code example contains one of the best tricks in the CMS. Use the type field with draws from several defined list definitions to make slight, or major changes, to the list display. The UI is open to configuration and innovation, and offers a good opportunity to use design skills. Anyone who can use HTML and CSS knows that a basic list can be turned into a float grid or be made inline. The combinations are limited only by what the designer can imagine and code.

In the code example, there is a custom logo field (u_logo) added to the Knowledge form. The custom field displays customer logos, partner logos, and award images on the awards page. There are several different sections that use this list definition so efficient reuse is taking place.

- div class="cms_knowledge_list customer_success" - Begin by creating an outer container with a unique class name that can be used as a basis for CSS style selectors and rules. From the outer container, many of the child elements can be accessed for theming.
- <g:for_each_record file="${current}" max="${jvar_max_entries}" > - Loop for list creation that calls the selected table record and the entries set on the list form.
- <img src="${current.u_logo.getDisplayValue()}
alt="${current.text}" width="110px"
/> - Defines linking to the article detail in the knowledge base. For further reference, look at content types within the site definition and notice some similarities. The knowledge.do? portion of the URL points to the knowledge detail page which is mandatory if you plan to call the knowledge base in your CMS site. The rest of the URL represents the syntax for calling a knowledge article by its sys_id. Each item housed within the system has a unique sys_id.
- <ttt>${SP}-
 ${SP}${current.author.first_name}${SP}${current.author.last_name}</ttt> - This line is commented out in the example. It has a jelly call ${SP} and pulls the author of the knowledge article by first and last name.
Style in Content Management

Content pages can be styled with CSS, just like any HTML website.

Three elements control CSS styles:

- **Style Sheets** are records containing CSS declarations.
- **Themes** are groups of style sheets that can be invoked together.
- **Frames** are UI macros that define the outer border of individual content blocks as they appear on a content page by calling on particular definitions in the style sheets.

Style sheets

Styles sheets are standard Cascading Style Sheets (CSS) that define the look and feel of all elements within the interface.

Cascading Style Sheets (CSS) can either be internal (stored in the database) or external (hosted on the server), based on organizational needs. To define an internal style sheet, use standard CSS in the style field. Using external CSS allows the Content Management System to use the same CSS as a corporate website or other online resource.

Use an external style sheet by defining a URL that points to the .cssx file. If you upload a .cssx file to the platform, the .cssx file can then be referenced using a URL.

Content pages do not reference style sheets directly. To invoke a style sheet, you must assign the style sheet to a content theme using the related list on the Content Theme form.
Content themes

Content themes are the convergence of structure and styling, making them a critical tool for creating a powerful user interface.

Understanding corporate design guidelines and maintaining communication with the art team responsible for the corporate website are important to the success of the project. If the organization has an art or design department that maintains branding, include it in this process.

A content theme is a collection of one or more style sheets (CSS files) that define a consistent look for a set of pages. In most environments, many pages share a few themes, usually one. Multiple themes can be used together within a site to create stylistic differentiators between site areas. A single content theme can create a unified look and feel for the site.
Note: Content themes are different from the color themes you can apply to your own system in System Settings.

Figure 570: Content Theme form

Customize a content theme
A theme is a collection of one or more style sheets (CSS files) that define a consistent look for a set of pages.

Role required: content_admin or admin
Themes can be invoked in any of the following ways.

- Directly by content pages.
- As the default theme of a content site.
- On the configuration page as a global default.

1. Navigate to Content Management > Design > Themes.
2. Click New.
3. Type a name for the theme and mark it as Active.
4. Right-click the form header and click Save. The Style Sheet related list appears.
5. Use the Style Sheet related list to add style sheets to the theme.

Doctypes

The view_content html page template (on which all CMS is based) defaults to doctype=html. The code looks like the following HTML source code.

```html
<!DOCTYPE HTML>
```

If your CMS site does not render properly, remove the doctype from the page by setting the following property:

- glide.html.doctype.pages =
  chat_desktop, live_feed, live_feed_small, navigator, navpage11, image_browse

The following is the default for this property.

- glide.html.doctype.pages =
  chat_desktop, live_feed, live_feed_small, navigator, navpage11, image_browse, view_content

Setting this doctype offers these benefits for building new sites:

- Incorporating common practice: Use a practice that is becoming widely adopted across the Internet and can prevent certain browsers from running in quirks mode.
- Cleaner CSS and markup: Write more standards-based CSS and markup to promote code sharing.
- A step towards browser compatibility: Find solutions that work across browsers and avoid browser-specific workarounds.

Format a frame

Frames provide a way to manage decorative containers for content blocks and any other elements within the site. For example, a frame can be a container (made of div or span tags) styled with rounded corners.

Role required: content_admin or admin

Individual content blocks use a frame UI macro to define the frames. When viewing a content block form, the Frame field offers a choice between the different frame UI macros. The frame UI macro does not, however, have the definition for the frame within its Jelly script. Instead, it references a particular frame as defined in a style sheet.

Configuring a new frame is a two-step process.

1. Define the frame in a style sheet.
2. Create the frame UI macro to invoke the frame definition.

Configure a frame in a style sheet

Role required: content_admin or admin

1. Navigate to Content Management > Design > Style Sheets.
2. Select a style sheet to contain the frame definition.
   Base system themes use a separate Frames style sheet.
3. Add the following code, substituting the desired frame name and style:

   ```html
   div.FRAMENAME{border:STYLE;}
   ```

4. Click **Update**.

   **Configure a frame UI macro**

   Role required: content_admin or admin

   1. Navigate to **Content Management > Design > Frames** and select one of the existing frame UI macros.
   2. Change the name to match the FRAMENAME you used in the style sheet.
   3. Right-click the header bar and select **Insert and Stay**.
   4. Update the frame name in the XML field as shown:

   ```xml
   <div class="FRAMENAME">
   ```

5. Click **Submit**.

   In any content block form, select the UI macro.

   **Content Management meta tags**

   Meta tags are special tags added to web pages that contain information about the page rather than being part of the page.

   Meta tags are not rendered and are not noticeable to a page visitor unless the visitor looks at the page source code. Web search engines read meta tags as they "crawl" the web, identifying and organizing content. Modern website designers often use meta tags to embed "hints" to search engines about how to index or otherwise crawl the site in question.

   You can define custom meta tags for content pages.

   Structurally, a meta tag consists of a tag a name/content pair and looks similar to this example.

   ```html
   <meta name="generator" content="MediaWiki 1.16wmf4" />
   ```

   CMS allows you to define both site level and page level tags.

   **Configure a page level meta tag**

   A page level tag is a meta tag defined on a specific page and included on only that page.

   Role required: content_admin or admin

   If an individual page has a specific tag with the same name as the site, the page tag takes precedence.

   1. Navigate to **Content Management > Pages**.
   2. Open the page.
   3. If not already included, add the **Meta Tags** related list to the form.
   4. In the **Meta Tags** related list, click **New**.
   5. Type a **Name** and **Content** for the tag.
   6. Click **Submit**.
Configure a site level meta tag

A site level tag is a meta tag defined on a site and included on every page within that site. If an individual page has a specific tag with the same name, the page tag takes precedence.

Role required: content_admin or admin

If an individual page has a specific tag with the same name as the site tag, the page tag takes precedence.

1. Navigate to **Content Management > Sites**.
2. Open the site.
3. If not already included, add the **Meta Tags** related list to the form.
4. In the **Meta Tags** related list, click **New**.
5. Enter a **Name** and **Content** for the tag.
6. Click **Submit**.

Configure DIV-based layouts

After you create your site, you can change the site layout with DIV tags.

Role required: content_admin or admin

Many web page layouts use tables for a consistent look and feel. Tables are effective for numbers and statistics, but can be limiting for designing other types of information. DIV tags are flexible block-element tags. To use DIV tags for layouts, give the tag an ID and assign attributes using CSS.

Changing to CSS and DIV tags help in the following ways:

- simplify code
- reduce the amount of code
- increase page load speed
- separate content from presentation
- help pages adapt to different device resolutions
- make pages easier for search engines to crawl
- make code more compliant with evolving web page design standards

A theme named **Administration Theme - Charcoal** is included as an example of CSS-driven, DIV-based layouts. To view the **Administration Theme - Charcoal** DIV-based layout, complete the following steps.

1. Navigate to **Content Management > Design > Layouts**.
2. Click **Admin 1 Column**
3. Review the code.

Content meta tag hierarchy

Page and site level meta tags are included in a content meta tag hierarchy.

**Site and page level tags**

CMS supports page level and site level tags. The system behavior is to show the sum of all page level and site level tags. For example, if you have two tags on the site and one tag on the page, all with different names, you see three tags on the page. Two are from the site and one is from the page. If both the site and the page have a tag of the same name, only the page tag is used.
Note: The Content Management System does not support some commonly used meta tags. For example, you cannot use the X-UA-Compatible meta tag, which allows you to specify the version of Internet Explorer for rendering a page.

**Example: all tags have unique names**

<table>
<thead>
<tr>
<th>Site Level</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>breakfast</td>
<td>oatmeal</td>
</tr>
<tr>
<td>lunch</td>
<td>sandwich</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Page Level</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>dinner</td>
<td>spaghetti</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>breakfast</td>
<td>oatmeal</td>
</tr>
<tr>
<td>lunch</td>
<td>sandwich</td>
</tr>
<tr>
<td>dinner</td>
<td>spaghetti</td>
</tr>
</tbody>
</table>

**Example: page level tag overrides site level tag**

<table>
<thead>
<tr>
<th>Site Level</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>breakfast</td>
<td>oatmeal</td>
</tr>
<tr>
<td>lunch</td>
<td>sandwich</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Page Level</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>lunch</td>
<td>salad &lt;--- overrides site level</td>
</tr>
<tr>
<td>dinner</td>
<td>spaghetti</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>breakfast</td>
<td>oatmeal</td>
</tr>
<tr>
<td>lunch</td>
<td>salad</td>
</tr>
<tr>
<td>dinner</td>
<td>spaghetti</td>
</tr>
</tbody>
</table>

**Content Management integration points**

Integration points use content blocks in CMS to link different applications together using static and dynamic methods.

By using integration points, users can connect to different systems from a single page. The power of the CMS is that it can display any data within the ServiceNow platform. The ESS portal, for example, connects users to a service catalog, a knowledge base, and a help and incident reporting site. Each section contains a set of links to additional content. List blocks offer the easiest way to display data dynamically. Content types define how lists link to the detailed data they reference.

Generating lists from the ServiceNow platform is straightforward, especially with filtering. When the CMS was first introduced, lists were the only method available to create navigational systems for sites. Content types define system record links using specific templates to define the list and the details.

- Summary template: defines the list
• Detail template: defines the detail as it is rendered in a page.

The templates work together to pass data attributes or *unique record identifiers* (sysid) into a single detail page that manages the data. Any attribute variations use the URL sent to the page.

Understanding how content types work can mean the difference between a site with 20 pages versus a site with 200 pages. There is a time for both types of data calls. Depending on the task, there are appropriate times to use static content and times for dynamic methods.

**View links between system elements and URLs**

You can view the links between your system elements and their URLs to render more specific content within your site.

Role required: content_admin or admin

See the *URL schema* on page 53 page for an overview of URL syntax in ServiceNow.

**Firefox Developer Tools:**

To understand what is being passed via the URL address bars between the system frame sets, use some available tools. For example, the Firefox browser has developer tools and an easy way to view records that render within the main content frame (gsft_main) of the ServiceNow system. Also, Firefox is good for quickly building menus and linking to records within the system. This Firefox functionality is useful when stepping through the *menu items* section on this page.

![Firefox Developer Tools](image)

**Figure 571: Frame view**

The Incidents list offers a quick example of viewing a form within its own tab. This viewing method illustrates how to append the URL string to render more specific content within your site.

1. Navigate to *Incident > All*.
2. Right-click in a blank area within the content frame and select *Open Frame in New Window* or *Open Frame in New Tab*.

The URL returned looks similar to this example.
<instance_name>/incident_list.do?
sysparm_userpref_module=b55b4ab0c0a80009007a9c0f03fb4da9

The URL comprises the following elements.

- **incident_list.do?** is the call to the list with no parameters passed into it. Explore context menu options and list filtering variations to see how the URL string is appended with each action.
- **incident_list.do?sysparm_query=&sysparm_view=ess** is an example of how the view is passed in the URL string.
- **incident_list.do?sysparm_query=GROUPBYactive&sysparm_view=** is how the URL looks when the **group by** option is chosen from the context menu.
- **incident_list.do?sysparm_query=category%3Dsoftware** is the result of applying a **category is software** filter to the list.

Explore all the options available within the right-click menu. Click through to the Incident form itself (incident.do) to see what the URL string does when parameters are passed into it.

**Element link examples**

View examples that show how to create a site pointing to various system data in several different data tables.

The examples illustrate setting up access to the following system data.

- Knowledge: building a versatile user interface for a knowledge base, from overview page to variations on the list definitions used in the site.
- Catalog: ideas for a business-to-consumer shopping experience that pulls various items and forms from your service catalog.
- Service Portfolio: using the business service portfolio to contain all defined services offered by your organization.
- Featured reports: methods for linking to the most important reports.
Figure 572: Example links to system data

Page source view

Use the tools available in your browser to view the frame source page and understand what URL address bars pass between system frame sets.

The Firefox browser, for example, has the developer tools option. This option provides an easy way to view records that render within the main content frame (gsft_main) of the ServiceNow platform. Also, Firefox quickly builds menus and links to records within the system.

This Firefox functionality is useful when stepping through the menu items section.
Static methods

In the CMS application shown, note the items highlighted in gray.

These CMS modules were created for ease of use. Form-based menu management and WYSIWYG code editing can be useful to both advanced and entry-levels users. The technical ability of subject matter experts (SMEs) managing the language of your site can vary considerably. Allowing SMEs to write the content and a technical resource to manage the linking is a good option. This division of labor can expedite menu and link creation within the system.
Navigation Menu Links are a great way to group similar links and are some of the most powerful interface components available to site designers using the CMS. They are base system templates used to group content links for placement on the page. Though the terminology is a bit different (menu sections and menu items), this type of linking behaves the same way as content links. For more information, see "Navigation menus and content links."

Static HTML Details are sometimes present on a page to which a link points. These content blocks are only useful for areas whose administrators are unfamiliar with HTML or markup. Administrators familiar with markup use dynamic blocks because they are extendable. For more information, see the section about static HTML on the "Using Content Blocks" page.

Content Links are the predecessors to navigation menus and were used when the CMS was first introduced. Use content links to make navigational links by defining several content links under the same category. Then, call the links as a list referencing the Content Link [content_link] table.

iFrame Details are used both dynamically and statically throughout the system. Using them can be an easy way to bring any form or list into your CMS pages. For more information, see the section about iFrames on the "Using Content Blocks" page.
Dynamic methods

While static methods are a powerful navigational tool, you have more control over data rendering using the dynamic methods.

Reference common code in the system to make long-term maintenance of the site easier. Coding skills are useful when implementing dynamic methods.

Figure 575: Static and dynamic methods

**Dynamic Blocks**

These blocks are where most of your work resides. For more information, see *Configure dynamic blocks* on page 2092.

**Frames**

Frames are meant to be decorative wrappers around any block in the system. They are mentioned with dynamic blocks because frames are essentially UI macros with the category of Frame. Frames apply the ${body} variable to a block and define where the block is inserted when rendered on the page.
Lists

Lists generate links to records based on the filtering rules you define. Lists can be sorted for presentation by any field in the corresponding record. Lists help supplement navigation and pull data from outside the CMS. For more information, see Customize a list block on page 2093.

Content Types

Content types provide site-specific control of how system data defined as templates is rendered. The rendering of lists is considered first (summary templates). The next consideration is the detail template, which allows control over the record rendering.

List Definitions

List definitions enable you to extend the summary template defaults defined on the site level. Used with frames, list definitions render the same data differently based on the placement on the page or site. For more information, see Configure list definitions on page 2095.

Detailed Content

Detailed content in page detail settings) are blocks that display the content of an existing document as a block on a content page. For more information, see Create a detailed content block on page 2072.

Example integration points

Each element on the page links to a specific URL point.

Figure 576: Integration points
**Business Services** links to a content page (CMS page referenced: Business Service Portfolio, URL: `business_service_category.do`) that pulls the system service catalog homepage into a frame within the content area. Each link within this section uses the browse by category page, where you pass in the name of the category to return results.

- **Target page iFrame URL**: `catalog_home.do?sysparm_nameofstack=aabdae07ef221000914304167b22567d&sysparm_view=business&sysparm_clear_stack=yes`  
  - **Target page frame name**: gsft_main
  - **Desktop Computing URL**: `category_browse.do?category=Desktop Computing`  
  - **Business Applications URL**: `category_browse.do?category=Business Applications`  
  - **Communications Services URL**: `category_browse.do?category=Communications Services`  
  - **Infrastructure Services URL**: `category_browse.do?category=Infrastructure Services`  
  - **Hosting Services URL**: `category_browse.do?category=Hosting Services`

**Featured Services** links to a content page which pulls a small subset of services into an iFrame.

- **iFrame URL**: `com.glideapp.servicecatalog_category_view.do?sysparm_parent=d67c446ec0a80165000335aa37eafbc1&sysparm_view=`  
  - **Frame name**: gsft_main
  - **Install Software URL**: `catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=10d69689c611227600ffeba41c664824`  
  - **Email Account URL**: `catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=d67a86b6c0a80165009386c752cd4a09`  
  - **Electronic Messaging URL**: `catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=533798810a0a0b2600f1a03593e19058`  
  - **VPN RSA Token URL**: `catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=d67b099ac0a80165019d0c276b772502`  
  - **Shared Storage (SAN) URL**: `catalog.do?uri=com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=cedd458a0a0b8300c3b1e32e7a3ac2`

**Reporting** links to a content page that pulls the reports page into an iFrame. All links within this menu refer to homepages in the system, which creates an issue with the home.do URL. Notice in the following links that `../` is used to create a relative URL outside of the CMS site home.do definition. Without this path, the site homepage would render within the iFrame.

- **iFrame URL**: `report_home.do`  
  - **Frame name**: gsft_main
  - **Cost Management Overview URL**: `../home.do?sysparm_userpref_homepage=fa81ae91c0a805c64c0942ab2e4b852b`  
  - **Administration Overview URL**: `../home.do?sysparm_userpref_homepage=8b7b11f6c611228901ff3fcfbd3cc8f`  
  - **Portfolio Overview URL**: `catalog_home.do?sysparm_view=business`  
  - **Service Availability URL**: `../home.do?sysparm_userpref_homepage=8ee772000a0abad00c38eb7e68b93d0`
• **Service Level Agreements (SLA)**
  URL: .../home.do?
sysparm_userpref_homepage=757e86a30a0006d4010a6851639498d1

Content Management testing

Test your site to ensure that all pages display correctly, links go to the specified address, and images are not broken. It is important to test the site as you build it. Do not wait until just before launch to begin testing.

Also, test templates as you create them so any issues are resolved before creating other content based on the templates. Recruit as many people as possible to help you test.

Here are some general site testing guidelines:

- Test on the browsers and platforms your site visitors use
- Test on various monitors (for example, LCD and CRT)
- View pages using different screen resolutions
- View pages using different color settings
- Test all navigation and links
- Test items that can be downloaded (for example, PDF files)
- Test the search functionality
- Test site security
- If necessary, test for accessibility (for guidelines, see the [W3C Website Accessibility Initiative](http://www.w3.org/))

Content Management administration

The Content Management System (CMS) is a tool for creating customized web pages and extending the look and feel of the ServiceNow platform.

Content Management administrators can use the **Content Management > Configuration** module to configure the CMS. Users with the content_admin role can configure CMS sites. Configuration options include setting a login page, selecting a default layout, and selecting a default theme. Page detail settings, content types, and macros can also be modified.

**Content configuration page**

Administrators set all global CMS defaults on the **Content Management > Configuration Page**. The options set on the configuration page can be overridden at a site level.

**Note:** The Content Management page cannot be domain separated.

**Table 509: Configuration page fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login page</td>
<td>Specifies a content page to use as the login page (if a login page is not set at the site level). Include fields on the login page where users can enter a user name and password to view the site.</td>
</tr>
</tbody>
</table>
### Field | Input value
--- | ---
Search page | Specifies a content page for displaying search results (if a search results page is not set at the site level).
Default layout | Specifies a default layout for content pages to use. For more information on layouts, see *Homepage and content page layouts* on page 494.
Default content theme | Specifies a default content theme for content pages to use. For more information, see *Content themes* on page 2106.

## Global search in Content Management

When you add global search to a CMS site, two different search result blocks can display, depending on the user role: global or no global.

Roles are defined at *System Properties > Global Text Search*.

- **Search Results (Global):** For users with permission to use the global search within the normal frame set. The normal frame set is defined as the default, non-CMS user interface with the set of frames.
- **Search Results (No Global):** For users without permission to use the global search. Searches only the knowledge base and the catalog.

The header search bar and the Search dynamic block in the base system both automatically handle the permissions and direct the user to the appropriate search results. In the base system, the search result blocks are deployed on the same Search Results content page.

In order for the global search to work properly, it is important to ensure that the DEFAULT directs users to a working content page. If not, the results from the global search link back to the frame set, not to pages within the CMS site.

**Note:** The global search is similar to the normal frame set user interface and ties to the same roles (set in *global text search properties*).

### Search Results (Global)

Before you use the **Search Results (Global)** dynamic block, define *content types* to control the behavior a search result is clicked. If no content type is defined for a table, selecting a link on that table renders the results according to the **DEFAULT** content type. It is good practice to point the **DEFAULT** content type to a page with a normal "Current Document" detailed block. Set the content type detail (**Detail Template** field) to have just an iFrame.

The **Search Results (No Global)** dynamic block does not require any additional configuration.

### CMS translation

You can translate CMS sites by activating internationalization plugins and manually translating custom interface strings.

Two tables support the translation of a CMS site into other languages.
• Translated Name / Field [sys_translated]: Stores strings that are shared or commonly used within a site. These include menu section names, menu item names, site breadcrumb names, link names, and footer menu links. Internationalization plugins typically provide translations for these strings. See Internationalization support on page 965.

• Translated Text [sys_translated_text]: Stores unique string translations which you create when you manually translate interface elements. See Translate the interface on page 991.

View a translated CMS site example

Activating an internationalization plugin provides a quick way to see translated strings for CMS menus, breadcrumbs, and links. For a full translation, you must translate the instance manually.

The following example explains how to view a translated site in Japanese.

1. Navigate to System Definition > Plugins.
2. In the Go to filter, select Name, enter I18N, and then press the Enter key.
3. Activate both the I18N - Japanese Translations and the I18N: Knowledge Management Internationalization Plugin v2 plugins.
4. Refresh the browser.
5. In the Language picker on the system toolbar, select Japanese.
6. Browse the site to see the translated language strings.

8. Browse the site to see the translated language strings.
Knowledge Management

The ServiceNow® Knowledge Management (KM) application enables the sharing of information in knowledge bases. These knowledge bases contain articles that provide users with information such as self-help, troubleshooting, and task resolution.

Knowledge Management has significantly changed with Knowledge v3, which is enabled by default for all instances starting with the Fuji release.

Knowledge Management supports processes for creating, categorizing, reviewing, and approving articles. Users can search and browse articles as well as provide feedback.

To support multiple groups, knowledge bases can be assigned to individual managers. Separate workflows can be used for publishing and retiring articles, and separate access controls can be used to control reading and contributing.

**Explore**
- Knowledge Management v3
- Knowledge homepage on page 2129
- Knowledge search on page 2131
- Contextual search
- Social Q&A on page 2132

**Set up**
- Video: Getting Started with Knowledge Management v3
- Knowledge Management setup guide for admins on page 2124
- Knowledge base setup guide for knowledge admins and managers on page 2126

**Administer**
- Select user criteria for a knowledge base on page 2143
- Video: How to Control Knowledge Access Through User Criteria
- Define a knowledge article category on page 2151

© 2017 ServiceNow. All rights reserved. 2123
Knowledge Management setup guide for admins

Before users in your organization can start creating knowledge bases and knowledge articles, you must set up Knowledge Management. Work with stakeholders to define requirements for setting up Knowledge Management effectively to meet the needs of users. For this task, you must be an administrator with the admin role.

Requirements

Role required
admin

Before you begin

Meet with the following stakeholders

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge manager</td>
<td>Defines and oversees the knowledge management processes for day-to-day operations related to content publishing and usage.</td>
</tr>
<tr>
<td>Knowledge admin</td>
<td>Configures advanced settings for specific Knowledge Management features</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>Responsibilities</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Content creators and reviewers</td>
<td>Create, review, and update articles in the knowledge bases.</td>
</tr>
<tr>
<td>Community managers</td>
<td>Focus on the creation and curating of social content.</td>
</tr>
<tr>
<td>Line managers</td>
<td>Manage teams that create and use knowledge articles</td>
</tr>
</tbody>
</table>

**With stakeholders, determine the following requirements**

- Who are the consumers of the content in the knowledge bases?
- Who are the writers, editors, and managers for the knowledge base articles?
- What are the required workflows to publish and retire articles?
- What are the unique behaviors required for your knowledge management implementation? For example, should attachments appear in search results?
- Do knowledge articles need to be translated into other languages?

**What to do**

**Assign knowledge roles to users and groups**
Determine which Knowledge Management roles on page 2128 are appropriate for each user, and then assign those roles.

**Create custom Knowledge workflows**
If the default knowledge workflows for publishing and retiring knowledge articles are not appropriate for knowledge bases, create custom workflows for those knowledge bases.

**Configure Knowledge properties**
If the default knowledge properties do not fulfill the requirements of users, reconfigure the properties as necessary.

**Activate support for other languages besides English**
If languages other than English must be supported, activate the Knowledge Management Internationalization plugin.

**Coordinate with knowledge managers for knowledge base setup**
Let knowledge managers and administrators know that they can start configuring their knowledge bases. For details, refer them to the Knowledge base setup guide for knowledge admins and managers on page 2126.
Next steps

Depending on any other requirements for the knowledge bases, you may be asked to help with additional tasks, such as configuring knowledge homepages.

Knowledge base setup guide for knowledge admins and managers

After basic Knowledge Management setup is completed, you can set up a knowledge base for users to create and publish knowledge articles.

Multiple knowledge bases can be created for different groups within your organization to share information within and between those groups.

Administrators create knowledge bases, and assign them to individual managers responsible for controlling the behavior and organization scheme of each knowledge base.

Each knowledge base contains knowledge articles that provide information for users, such as policy, release notes, or instructions for a task. Each knowledge base can also use a separate workflow for publishing and retiring articles.

Requirements

Role required

knowledge_administrator, knowledge_manager, or admin

Before you begin

Determine the following requirements for each knowledge base that you want to set up:

• Who are the knowledge base managers that are responsible for approving articles?
• Who are the users and contributors for that knowledge base? Access for these users is defined through user criteria.
• What are the categories to be used to classify articles? Will users be able to create categories?
• Will users of the knowledge base be able to create new content?
• Will Social Q&A be leveraged for this knowledge base?

What to do

Set up the knowledge base

1. Create the knowledge base.
2. Set up which users can read, create, or edit knowledge articles by selecting user criteria for the knowledge base.
3. Define the approval process for articles using workflows.
4. Define the knowledge categories that are needed for organizing articles in the knowledge base.

5. If you want to enable users to ask and answer questions in the knowledge base, activate Social Q&A on page 2132.

6. If you want a custom knowledge homepage, ask your ServiceNow administrator to create it.

Next steps

Let users in your organization know that they can start creating and searching articles in the knowledge base. For details, refer users to the Knowledge Management guide for users on page 2127.

If you have the knowledge_manager role, there are many other tasks that you can perform to maintain the knowledge base. You can pin articles so they are featured prominently in the search results and on homepages. You can also assign other users as managers of a knowledge base. For details, see Knowledge manager on page 2143.

Knowledge Management guide for users

After knowledge bases are set up, you can start searching and creating articles. If Social Q&A was activated for the knowledge base, you can also submit answers or answer questions for other users.

Requirements

Role required
None. All users can read, create, and edit knowledge articles in a knowledge base, unless the knowledge manager has restricted access to it.

If you need these permissions, contact the knowledge manager for the knowledge base.

What to do

Create knowledge articles
If you have the required permissions for a knowledge base, you can create articles for it. You can create articles from the Self-Service application menu or by importing Word documents.
You can also create knowledge articles from incidents and problems.

Search for knowledge articles
From the Knowledge homepage on page 2129, you can select a knowledge base to search for articles and answers. You can view only the knowledge bases to which you have access.
You can also search knowledge from within an incident.
Access to knowledge using your mobile device is supported.

**Ask and answer questions**

If Social Q&A on page 2132 was activated for a knowledge base, you can ask and answer questions. You can also browse and vote on answers.

**Next steps**

After articles are created, you can perform the following tasks to make sure that articles are organized in the right knowledge bases and retired when appropriate. If existing knowledge bases do not fit your needs, you can request a new knowledge base.

- Move a knowledge article on page 2140
- Retire a knowledge article on page 2141
- Request a knowledge base on page 2138

**Knowledge Management roles**

Certain roles are required to use Knowledge Management functionality.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any role</td>
<td>All users with at least one ServiceNow role can read, create, and edit knowledge articles. However, the knowledge manager of a knowledge base can select criteria to restrict this access.</td>
</tr>
<tr>
<td>knowledge</td>
<td>Users with the knowledge role can contribute to the default knowledge base and access the Knowledge application menu.</td>
</tr>
<tr>
<td>knowledge_manager</td>
<td>Knowledge managers perform administrative functions for knowledge bases they manage such as defining categories, pinning important articles, and approving changes to articles. Users selected as managers of a knowledge base receive this role automatically.</td>
</tr>
<tr>
<td>knowledge_admin</td>
<td>Knowledge administrators can create knowledge bases and manage the default knowledge base.</td>
</tr>
<tr>
<td>admin</td>
<td>Administrators can configure knowledge workflows, set knowledge properties, and manage knowledge forms and homepages.</td>
</tr>
</tbody>
</table>

Administrators assign these roles to the users and groups who maintain the knowledge content.
Knowledge homepage

The Knowledge homepage displays knowledge articles and social Q&A questions organized by knowledge base and category, as well as featured content and popular articles.

To view the default knowledge homepage navigate to **Self-Service > Knowledge**.

**Note:** The knowledge homepage is not compatible with Internet Explorer 9 or earlier. Using one of these browsers will cause you to be redirected to the legacy knowledge portal.

![Knowledge homepage screenshot](image)

**Figure 577: The Knowledge homepage**

If you access knowledge from a service management application, the knowledge homepage for the associated application opens.

From the homepage, you can import a Word document to a knowledge base using the **Import** button. You can also and create a new article, or ask a question using the **Create new** button.
• Import a Word document
• Create a new article
• Ask a question

Note: You must have user criteria “Can contribute” permission for at least one active knowledge base, otherwise the Import and Create new buttons do not appear. See Knowledge Management v3 migration on page 2169 and Knowledge manager on page 2143.

You can select a knowledge base to browse articles and questions within that knowledge base. You can view only knowledge bases you can access.

Figure 578: Browsing articles by category

Articles and questions are organized by category or by tag. Categories are listed alphabetically. While browsing, you can filter content by type to view only knowledge articles, only social Q&A questions, or only unanswered questions. You can sort content by most recent update or by number of views.

An administrator can configure the knowledge homepage to display the number of articles and questions within each category. This count includes articles and questions from subcategories. To display the article and question count, set the knowledge homepage property Display or hide the count of articles and questions in the category and child categories (glide.knowman.show_number_on_categories) to true.
Knowledge search

You can search for knowledge articles and social Q&A questions from the knowledge homepage using the search bar at the top of the page.

Search results include only articles and questions you are authorized to read. Documents that are attached to articles are also listed in the search results (for those articles to which you have access).

**Note:** To change how the attachments are displayed in the search results, set the *How to display attachments in Knowledge Search Results* property. Attachments can be displayed with a snippet, link only, or not at all.

You can sort knowledge content by relevancy, most recent update, or number of views.

You can filter results using the check boxes that appear. Filtering options appear depending on the search text.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Select if you want to view knowledge articles, social Q&amp;A questions, or both. This option is only available if Social Q&amp;A is enabled.</td>
</tr>
<tr>
<td>Knowledge Bases</td>
<td>Select a knowledge base to search. You can select only knowledge bases you can access. If you do not select a specific knowledge base, search results include articles and questions from all knowledge bases that you can access. You can also select a knowledge base from the choice list in the search bar.</td>
</tr>
<tr>
<td>Categories</td>
<td>Select one or more knowledge categories. Categories are listed alphabetically.</td>
</tr>
<tr>
<td>Authors</td>
<td>Select one or more authors.</td>
</tr>
</tbody>
</table>
Social Q&A

If Social Q&A is activated for a knowledge base, you can ask questions and respond to questions from other users. You can also vote on helpful questions and answers.

Social Q&A extends the Knowledge application. All questions and answers are associated with a knowledge base. Social Q&A uses Knowledge access controls, search, and the knowledge homepage.

Social Q&A is available on mobile devices using the mobile knowledge interface. You can perform all Social Q&A functions on mobile, such as asking and answering questions.

**Note:** Social Q&A is not compatible with Internet Explorer 9 or earlier. If you navigate to a Social Q&A page with one of these browsers, a browser support warning appears on the page.
Enable social Q&A for a knowledge base

As a knowledge manager, you can enable social Q&A for a knowledge base you manage.

Role required: knowledge_manager

All Social Q&A questions and answers are associated with a knowledge base. The **Enable social questions and answers** check box on the Knowledge Base form controls if users can view and ask questions.

Social Q&A uses user criteria from the knowledge base to determine which users have access to questions. A user must meet the criteria defined in the knowledge base **Can Read** related list to view or contribute to Social Q&A questions, answers, and comments associated with that knowledge base.

**Note:** The **Can Contribute** related list does not control the ability to contribute to Social Q&A. Users that meet the **Can Read** criteria for a knowledge base can also contribute questions, answers, and comments.

1. Navigate to **Knowledge > Knowledge Bases**.
2. Select a knowledge base you manage.
3. Select the **Enable social questions and answers** check box.
   - This check box only appears if the Social Q&A plugin has been activated.
4. Click **Update**.

You can disable Q&A for a knowledge base by clearing the **Enable social questions and answers** check box. Disabling Q&A does not delete existing questions and answers associated with this knowledge base, but prevents users from browsing or searching for those questions and answers.

Social Q&A questions

Social Q&A organizes information by question.

You can browse and search for questions from the Knowledge homepage (**Self-Service > Knowledge**). Questions appear along with knowledge articles organized by knowledge base, category, and tags.

Click on a question to view the question details, as well as responses and comments. You can add responses and comments, vote on existing responses, share questions, and edit your own questions and answers from the question details.

Ask a question

As a Social Q&A user, you can ask questions that other users can respond to.

Role required: none

1. Navigate to **Self-Service > Knowledge**.
2. Click **Create new**.
3. Click **Question**.
4. Enter a value for the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Enter the question you have.</td>
</tr>
</tbody>
</table>
### Field | Description
---|---
Question details | Enter additional details about the question that may help other users provide an answer.
Knowledge base | Select the knowledge base this question relates to. You can select only knowledge bases configured to allow Q&A.
Category | Select the knowledge category this question relates to. You can select only categories within the selected knowledge base. Questions without a category appear on the knowledge homepage in the (empty) category.
Tags | Enter one or more tags that describe the question.

5. Click **Post Question**.

To accept the answer, click the **Accept** link from within the answer. The accepted answer moves to the top of the list of answers. You can unaccept an answer by clicking **Unaccept**.

**Note:** You must be the owner of the question or the knowledge manager to accept an answer.

### Answer a question

As a user, you can answer a question another used has asked. The owner of the question or the knowledge manager can then accept the answer.

**Role required:** none

As a knowledge manager or the owner of a question, you can accept an answer as the correct answer. That answer then appears above other answers for the question.

1. Navigate to **Self-Service > Knowledge**.
2. Select a question.
3. Enter an answer in the **Your answer** field.
4. Click **Submit**.

### Comment on a question or answer

You can comment on a question or an answer to provide additional information relevant to that question or answer.

**Role required:** none

1. Navigate to **Self-Service > Knowledge**.
2. Select a question.
3. Within the question or answer you want to comment on, click the **Comment** link.
4. Enter the comment text.
   - Comments have a maximum length of 140 characters.
5. Click **Post comment** or press the Enter key.
Edit a question, answer, or comment

You can edit questions, answers, and comments that you submit, or in knowledge bases you manage.

Role required: none

1. Navigate to Self-Service > Knowledge.
2. Select a question.
3. Within the question, answer, or comment you want to edit, click Edit.
4. Edit the details as needed.
5. Optional: When editing a question, click Advanced to modify the question record.
6. Click Update.
   Click Undo to discard the changes and close the editor.

Vote on a question or answer

As a Social Q&A user, you can vote on a question or answer to promote it.

Role required: none

Vote on questions that you want users to answer, and answers that you believe accurately resolve questions. Alternatively, vote down questions that you do not find useful, or answers that you believe are incorrect. You can vote for each question or answer only once, but you can change your vote.

Answers with a higher score appear above answers with a lower score when viewing a question. A pinned answer appears above other answers regardless of votes.

   Note: You cannot vote for your own questions or answers.

1. Navigate to Self-Service > Knowledge.
2. Select a question.
3. Click the up or down arrow next to the question or answer you want to vote on.
   The score for that question or answer changes depending on your vote.

You can change your vote by clicking the other arrow.

Subscribe to a question

You are automatically subscribed to any question you ask, so you receive notifications when another user votes on, comments on, or answers the question. If you want to receive notifications for a question asked by another use, you can manually subscribe to the question.

Role required: none

1. Navigate to Self-Service > Knowledge.
2. Select a question.
3. Click the subscribe icon.

If you want later to stop receiving notifications about this question, click the unsubscribe icon.

Delete a question, answer, or comment

You can delete questions, answers, or comments that you submitted.
You can delete only questions, answers, and comments you submitted. Knowledge managers can delete any question, answer, or comment within knowledge bases they manage.

Role required: none

1. Navigate to **Self-Service > Knowledge**.
2. Select a question or answer.
3. Within the question, answer, or comment you want to delete, click **Delete**.

Deleting a question also deletes all answers and comments associated with that question. Deleting an answer also deletes all comments associated with that answer.

Share a question

You can generate a URL directly to a question and related answers.

Role required: none

1. Navigate to **Self-Service > Knowledge**.
2. Select the question that you want to share.
3. Click **Share**.

Some browsers do not allow adding content directly to the clipboard. You may need to manually copy the URL from the window that appears.

Distribute the URL to share the question.

Social Q&A internationalization support

The Social Q&A questions and answers are automatically associated with the language that is set for the interface.

You can only browse and search for questions in the language currently set for the interface. For example, if a Social Q&A user has set French as the interface language, any questions that the user asks is associated with the French language. All Social Q&A users who want to view those questions must have their interface language set to French.

**Note:** For Social Q&A internationalization to take effect, you must clear the platform cache and browser cache after selecting the desired language.

Social Q&A search

Users can search for specific questions from the knowledge homepage.

Social Q&A uses knowledge search to provide search results. Searching knowledge returns questions that contain the search terms in the question title or the question details, or in the accepted answer.

By default, knowledge search results include both knowledge articles and questions. You can filter results using knowledge search controls, such as filtering by knowledge base or author.

Social Q&A results also appear with knowledge articles when using global search.

Social Q&A uses the Contextual Search feature to generate search results. Questions appear in Contextual Search results, such as when creating an incident. The Social QA Question Search Context and Social QA Question Searcher records define the Social Q&A search. To ensure Social Q&A search functions as intended, do not modify these records.

Social Q&A tags

Tags organize questions and provide information about the question subject matter.
Social Q&A shares available tags with other applications allowing you to organize records consistently across the instance. Tags are stored on the Tags [label] table.

You can view a list of tags and filter by tag from the knowledge homepage. Clicking on a tag displays a list of knowledge articles and questions with that tag.

Only users who can edit a question can edit the associated tags. By default, only the question owner and knowledge managers can add or remove tags on questions.

Knowledge feedback

You can view and contribute to feedback on knowledge articles.

Feedback options

You can submit feedback for knowledge articles in these ways:

- Flag an article as incorrect or inappropriate.
- Provide a rating value for the article.
- Mark an article as helpful or not helpful.
- View comments, add a new comment, or reply to existing comments.

Users can view comments directly on the article. Knowledge managers can view the other types of feedback by navigating to Knowledge > Feedback.

Administrators and knowledge managers can disable some feedback options using fields on the Knowledge Base form when creating a knowledge base. Administrators can configure feedback options using knowledge properties.

Flagging articles

You can flag an article for incorrect or inappropriate content. Click Flag Article in the article header to open a new window, allowing you to enter suggested changes.

Suggestions you make when flagging an article do not appear in the public comments for that article.

Rating articles

The five stars below the article title allow you to indicate the article's effectiveness on a scale of 1 to 5.

Marking articles

The question Helpful? at the bottom of the article allows you to indicate the usefulness of the article with a simple Yes or No answer.

Comments

Knowledge comments at the bottom of the article use live feed to enable a conversation around a knowledge article. For example, you can post replies to comments, add attachments, or Like comments.
Request a knowledge base

If existing knowledge bases do not fit your needs, you can request a new knowledge base through the service catalog. If the request is approved, you are added as the owner of the new knowledge base.

Role required: none

1. Navigate to Self-Service > Service Catalog.
2. Select Can We Help You?.
3. Select the Request Knowledge Base catalog item.
4. Enter the reason you want the knowledge base, and a name for the knowledge base.
5. Click Submit.

You are notified when the request is approved or rejected. If the request is approved, you are added as the owner of the new knowledge base. You can then assign managers and manage (pin) articles in the new knowledge base.

Knowledge bases created through this request process are inactive by default, so you must activate the knowledge base to make it available for users.

Create a knowledge article

Knowledge contributors can create and edit knowledge articles within a knowledge base to share information across your organization.

Users with at least one role can create and edit knowledge. These users are known as knowledge contributors. Users without any role can read articles and submit feedback, but cannot create or edit articles.

Some knowledge bases may allow only certain users to contribute. For example, a member of the IT department can create knowledge articles in the IT knowledge base, such as desktop support information or articles describing company IT processes.

1. Navigate to Self-Service > Knowledge.
2. Click Create new, then click Article.

Note: You must have user criteria "Can contribute" permission for at least one active knowledge base, otherwise the Create new button does not appear. See Knowledge Management v3 migration on page 2169 and Knowledge manager on page 2143.

3. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Base</td>
<td>The knowledge base selected for this article.</td>
</tr>
<tr>
<td>Category</td>
<td>The category for this article. Select a Knowledge Base before you select a category. Articles without a category appear on the knowledge homepage in the (empty) category.</td>
</tr>
<tr>
<td>Published</td>
<td>When this knowledge article was published. This value is set when the article is created, and updated when the article is published.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Valid to</td>
<td>When this knowledge article expires. Articles do not appear in search results after the valid to date, or if the valid to date is empty.</td>
</tr>
<tr>
<td>Image</td>
<td>An image that appears beside the article when searching from the legacy knowledge portal.</td>
</tr>
<tr>
<td>Workflow</td>
<td>[Read-Only] The publication state of the article, such as Draft or Published. When inserting a new article from an existing article, the state of the new article is reset to Draft.</td>
</tr>
<tr>
<td>Source</td>
<td>The task this knowledge article was created in response to, if any. This field is set automatically when you create the knowledge article from a task record.</td>
</tr>
<tr>
<td>Attachment link</td>
<td>Check box for downloading an attached file automatically when a user accesses the article, instead of opening the article view. Add one or more attachments to the article to use this option. <strong>Note:</strong> You may attach multiple files, but most web browsers permit users to download only the first one. To ensure download of all the files, bundle them into an archive, such as with WinZip, and attach the archive.</td>
</tr>
<tr>
<td>Display attachments</td>
<td>Check box for displaying attachments to users viewing this knowledge article. Attachments appear below the article text. Add one or more attachments to the article to use this option.</td>
</tr>
<tr>
<td>Short description</td>
<td>The title of the article. This title appears when browsing and searching knowledge, and at the top of the article.</td>
</tr>
<tr>
<td>Text</td>
<td>Content for the article. Use the WYSIWYG HTML editor to create content. A preview of the content appears when browsing and searching knowledge.</td>
</tr>
</tbody>
</table>

4. Click **Submit** to create the article.

After saving the article record, you can add tags to further organize the article. Any additional steps required to publish the article, such as approvals, depend on the publishing workflow for the knowledge base.
Select a knowledge article category

Knowledge articles within a knowledge base are grouped by category. These groups can help you define the knowledge base taxonomy, and can help users find articles within that knowledge base.

As a knowledge contributor, when editing a knowledge article you can select categories using the Category picker, and add or edit categories if enabled for the knowledge base.

1. Click the reference lookup icon beside the **Category** field to open the Category picker.

![Category picker](image)

2. Select an existing category, or click the add category icon (+) to add a new category.

   After you select a category, you can click the pen icon to rename that category. Press the enter key or click outside the selected category to save the change.

   **Note:** Clicking **Cancel** while selecting a category cancels selecting a new category, but does not revert any changes you make to the categories such as renaming or adding new categories.

3. Fill in the fields on the form, as appropriate.
4. Optional: Select or add a subcategory.
5. Click **OK**.

Move a knowledge article

You can move articles between knowledge bases.

1. Edit a knowledge article.
2. Change the **Knowledge Base** field value.

   After moving an article, the Apply Default Values dialog box prompts you to populate certain fields in the article with default values from the new knowledge base. If the new knowledge base has no default values, the dialog box does not appear.

3. Click **OK** to overwrite fields with the default values, or **Cancel** to leave all fields unchanged.

   A knowledge manager can define default values for articles in knowledge bases you manage using the **Set default knowledge field values** field on the Knowledge Base form.
Retire a knowledge article

Retired knowledge articles are no longer available for users to view.

A knowledge article has an associated retirement workflow, similar to the publishing workflow. This allows administrators to configure these workflows, defining an approval and review process for retiring knowledge if appropriate.

When editing an article, click Retire to launch the retirement workflow associated with that article.

Create knowledge from an incident or problem

To create knowledge from an incident or problem, select the Knowledge check box on the incident or problem form and close the incident or problem record.

The short description from the incident or problem becomes the knowledge article title. Articles created this way are added to the knowledge base specified in the property glide.knowman.task_kb.

Depending on the value of the glide.knowman.submission.workflow property, knowledge created from an incident or problem may require additional approval. When this property is true, a submission record is created instead of a knowledge article. A user with the knowledge role must approve the submission to create a knowledge article.

Approving a submission creates a new knowledge article using the submission content.

1. Navigate to Knowledge > Open Submissions.
2. Select a submission record.
3. Review the submission to ensure the content and settings are correct.
4. Click Create Article.

A new draft knowledge article is created. Articles created this way are added to the knowledge base specified in the property glide.knowman.task_kb.

You can edit the new knowledge article before publishing it, such as to select a category.

Import a Word document to a knowledge base

Import a Microsoft Word document to create a new knowledge article.

Role required: None. You must have user criteria “Can contribute” permission for at least one active knowledge base. If you cannot contribute, the Import button does not appear.

Note: An administrator can control the visibility of the Import Articles button by setting the other knowledge property Hide the 'Import' functionality (button and drag-n-drop) for all users (glide.knowman.import.hide_import_functionality) to true.

Importing documents to knowledge is available only from a computer browser. You cannot import documents from a mobile device.

You can import Microsoft Word .doc and .docx files.

Note: The file extension must be lowercase.

When you import a document, text content from the document is used to create the knowledge article. All HTML supported by the TinyMCE editor such as tables, lists, and links, as well as styling such as bold and
italics, are preserved. Images from the document are added as attachments to the knowledge article and embedded in the article body.

**Note:** You can import multiple files at a time. Closing the browser or navigating away cancels any in-progress uploads but does not delete articles created from completed imports.

You cannot import documents to knowledge from mobile devices.

The following styles and elements are preserved when importing a .doc or .docx file into a knowledge base. Styles and elements not included in this list may not be preserved when importing a document.

- Titles
- Headings
- Images
- Links
- Bold text
- Italic text
- Underlined text
- Ordered and unordered lists
- Tables

**Note:** Only default heading settings are supported. Custom heading styles are imported using the default settings for those styles. Table styling and borders are not supported.

1. Navigate to **Self-Service > Knowledge**.
2. Perform one of the following actions.
   - Click **Import** and select one or more document files.
   - Drag one or more document files onto the knowledge homepage.
3. Enter a **Short description** for each knowledge article.
   The default value is the document filename without the file extension.
4. Select the **Knowledge base** to add the new article to.
   You can select only knowledge bases you can contribute to.
5. Optional: Select a **Category** from within that knowledge base.
6. Optional: Select the **Publish** check box to start the publishing workflow for each imported article immediately after the import finishes.
   This check box applies only when importing to v3 knowledge bases and only if the **Show publish checkbox on the knowledge import pop-up** property is set to **Yes** on the Knowledge Management properties form. You cannot automatically publish articles imported to v2 knowledge bases.
   Knowledge v2 articles go into **Draft** state. Knowledge v3 articles go into **Review** or **Published** state, depending on the workflow attached to the knowledge base (Approval Publish or Instant Publish).
7. Click **Import**.

A new article is created in the selected knowledge base and category using the content from the uploaded document. If you uploaded multiple documents, one article is created for each. If you use knowledge internationalization, the language of the new article is set to the currently selected system language.

After the upload completes, a popup window appears displaying the number and short description of the newly-created articles. Click an article to view the full record.

If any errors occur during the upload, a popup window appears to display the error.
Knowledge manager

Having multiple knowledge bases allows an organization to spread management responsibilities across multiple users, known as knowledge managers.

As a knowledge manager, you can assign other managers, define category structures, configure which users can read and contribute articles, move and pin articles, and modify most fields on the Knowledge Base form. You can also approve the publishing or retiring of knowledge articles in those knowledge bases. You can enable Social Q&A for your knowledge bases if the Social Q&A application is active.

The primary manager of a knowledge base is the owner of that knowledge base. Each knowledge base must have one owner. There may be any number of additional managers for each knowledge base. All managers of a knowledge base, including the owner, automatically receive the knowledge_manager role.

Pin a knowledge article

You can pin an article to appear at the top of knowledge search results and in the Featured content section of the knowledge homepage.

Pinning associates an article to specific keywords. Searching for a keyword causes articles with that keyword to appear at the top of search results. You can pin articles that need to be distributed broadly, such as a maintenance notice or new HR policy.

1. Navigate to Knowledge > Knowledge Bases.
2. Select a knowledge base you manage.
3. In the Featured content related list, click New.
4. Select a Knowledge article
5. Click the lock icon to expand the Keywords field.
6. Select or create knowledge keywords using the reference lookup icon.

**Note:** Only articles with the keyword homepage are shown in the Featured Content section of the knowledge homepage.

7. After adding all keywords, click Submit.

Select user criteria for a knowledge base

You can specify user criteria to control which users can create, read, write, and retire knowledge articles.

User criteria refers to knowledge base records that determine the users who can read or contribute to a knowledge base. User criteria definitions:

- **canRead** users who can read all knowledge base articles
- **cantRead** users who cannot read, create, or modify articles in the knowledge base
- **canContribute** users who can read, create, and modify articles in the knowledge base
- **cantContribute** users who cannot create or modify articles in the knowledge base

**No User Criteria**

If a knowledge base has no user criteria, all users can read its articles. Any user having at least one role can create and edit articles, unless they are members of the cantContribute user criteria. If a knowledge base has canRead user criteria, but no canContribute user criteria, all users with at least one role can
access and modify the knowledge base. This video demonstrates how to control Knowledge Management access through user criteria.

1. Navigate to **Knowledge > Knowledge Bases**.
2. Select a knowledge base you manage.
3. Access one of these related lists, and then select or create user criteria records.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can read</td>
<td>Users who meet one of these criteria can read articles in this knowledge base.</td>
</tr>
<tr>
<td>Can contribute</td>
<td>Users who meet one of these criteria can create and edit articles in this knowledge base.</td>
</tr>
</tbody>
</table>
Can Read
Leveraging User Criteria to grant read access to Knowledge Bases & Knowledge Articles

User role: admin, knowledge_admin, KB-owner, KB-manager, CanContribute

No

Yes

User has Blacklisted role

No

User has Whitelisted role, or KB requires no role

No

Can Read KBs and/or Article(s)
Can Contribute

Leveraging User Criteria to grant access to Knowledge Bases & Knowledge Articles

User role: admin, knowledge_admin, KB-owner, KB-manager

No

User has Blacklisted role

Yes → No Access to KB and/or Article(s)

Yes → No

User has Whitelisted role, or KB requires no role

No

Can Contribute to KB and/or Article(s)
Table 514: Order of System Validation (L->R)

<table>
<thead>
<tr>
<th>Contribute</th>
<th>Read</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>cantContribute (1)</td>
<td>canContribute (2)</td>
<td>cantRead (3)</td>
</tr>
<tr>
<td>empty</td>
<td>empty</td>
<td>empty</td>
</tr>
<tr>
<td>empty</td>
<td>empty</td>
<td>empty</td>
</tr>
<tr>
<td>empty</td>
<td>empty</td>
<td>User B</td>
</tr>
<tr>
<td>empty</td>
<td>empty</td>
<td>User B</td>
</tr>
<tr>
<td>empty</td>
<td>empty</td>
<td>User B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contribute</th>
<th>Read</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>cantContribute (1)</td>
<td>canContribute (2)</td>
<td>cantRead (3)</td>
</tr>
<tr>
<td>empty</td>
<td>User C</td>
<td>empty</td>
</tr>
<tr>
<td>empty</td>
<td>User C</td>
<td>empty</td>
</tr>
<tr>
<td>Contribute</td>
<td>Read</td>
<td>Result</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>cantContribute (1)</td>
<td>canContribute (2)</td>
<td>cantRead (3)</td>
</tr>
<tr>
<td>empty</td>
<td>User C</td>
<td>empty</td>
</tr>
<tr>
<td>empty</td>
<td>User C</td>
<td>User B</td>
</tr>
<tr>
<td>User D</td>
<td>empty</td>
<td>empty</td>
</tr>
<tr>
<td>Contribute</td>
<td>Read</td>
<td>Result</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>cantContribute (1)</td>
<td>cantRead (3)</td>
<td>canRead (4)</td>
</tr>
<tr>
<td>User D</td>
<td>empty</td>
<td>empty</td>
</tr>
</tbody>
</table>
| | | | 1. User D -> Denied Create, Read
| | | | 2. All Other Users with role -> Create
| | | | 3. User A -> Read
| | | | 4. All Users with role -> Read
| User D | empty | User B | empty |
| | | | 1. User D -> Denied Create
| | | | 2. All Other Users with role -> Create
| | | | 3. All Users -> Read
| User D | empty | User B | User A |
| | | | 1. User D -> Denied Create, Read
| | | | 2. All Other Users with role -> Create
| | | | 3. User A -> Read
| | | | 4. All Users with role -> Read
<table>
<thead>
<tr>
<th>Contribute</th>
<th>Read</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>cantContribute (1)</td>
<td>cantRead (3)</td>
<td>canRead (4)</td>
</tr>
<tr>
<td>User D</td>
<td>User C</td>
<td>empty</td>
</tr>
<tr>
<td>User D</td>
<td>User C</td>
<td>empty</td>
</tr>
<tr>
<td>User D</td>
<td>User C</td>
<td>User B</td>
</tr>
<tr>
<td>User D</td>
<td>User C</td>
<td>User B</td>
</tr>
<tr>
<td>User D</td>
<td>User C</td>
<td>User B</td>
</tr>
<tr>
<td>User D</td>
<td>User C</td>
<td>User B</td>
</tr>
<tr>
<td>User D</td>
<td>User C</td>
<td>User B</td>
</tr>
</tbody>
</table>

**Note:** In the tables of rules, an implicit last rule is: All Users -> Denied Create, Read.
Define a knowledge article category

Each knowledge base has a hierarchy of categories that organize the articles.

A category can be a top-level category or a child category. The parent of a child category can be a top-level category or another child category. By making a category the child of another child category you can create a category structure with any number of levels.

You can create and edit categories separately for each knowledge base.

1. Navigate to Knowledge > Administration > Knowledge Bases.
2. Select a knowledge base you manage.
3. Choose an option.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a top-level category.</td>
<td>Navigate to the Knowledge Categories related list and click New.</td>
</tr>
<tr>
<td>Create a child category.</td>
<td>Expand an existing category and click New within the expanded section.</td>
</tr>
</tbody>
</table>
4. Enter a **Label** for the category. The label appears as the name of the category.
5. Optional: Enter a **Value** you can use when referencing the category in scripts.

6. Click **Submit**.

   In addition to using categories defined by a knowledge manager, knowledge contributors can add and edit categories when editing an article.

**Assign a knowledge base manager**

You can assign users as managers of a knowledge base.

1. Navigate to **Knowledge > Knowledge Bases**.
2. Select a knowledge base you manage.
3. In the **Managers** field, click the unlock icon.
4. Add one or more users.
5. Click **Update**.

   After you save the knowledge base record, users selected as managers automatically receive the knowledge_manager role if they do not have it already.

You can remove a knowledge base manager by removing that user from the **Managers** field. If the user is not a manager of any other knowledge base, the knowledge_manager role is removed from that user.

**Knowledge administration**

As an administrator, you are responsible for setting up knowledge bases, configuring knowledge workflows, and customizing knowledge homepages. Most day-to-day management functions are performed by knowledge managers.

The following podcast offers additional information on using the Knowledge Base Application.

**Create a knowledge base**

An administrator can set knowledge base field values when creating a knowledge base. Knowledge managers can set field values for knowledge bases they manage.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>The name of the knowledge base.</td>
</tr>
<tr>
<td>Icon</td>
<td>The image that is displayed next to an article from this knowledge base on the search and browse result pages.</td>
</tr>
<tr>
<td>Disable commenting</td>
<td>A check box for preventing users from commenting on articles in the knowledge base. You can override this setting for specific articles using the <strong>Disable commenting</strong> check box on the Knowledge form.</td>
</tr>
<tr>
<td>Disable suggesting</td>
<td>A check box for preventing users from suggesting edits to articles in the knowledge base. You can override this setting for specific articles using the <strong>Disable suggesting</strong> check box on the Knowledge form.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Disable category editing</td>
<td>A check box for preventing contributors from creating and editing categories when selecting a category. When this check box is selected, only knowledge managers can define knowledge categories.</td>
</tr>
<tr>
<td>Owner</td>
<td>The user responsible for the knowledge base.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Knowledge managers cannot change the <strong>Owner</strong> field value.</td>
</tr>
<tr>
<td>Managers</td>
<td>A list of knowledge base managers.</td>
</tr>
<tr>
<td>Publish workflow</td>
<td>The publishing workflow for articles within the knowledge base.</td>
</tr>
<tr>
<td>Retire workflow</td>
<td>The retiring workflow for articles within the knowledge base.</td>
</tr>
<tr>
<td>Active</td>
<td>A check box indicates whether the knowledge base is active. When the box is not checked, only users with the admin role can create or view its articles.</td>
</tr>
<tr>
<td>Enable social questions and answers</td>
<td>A check box indicates whether the knowledge base allows Social Q&amp;A questions.</td>
</tr>
<tr>
<td>Description</td>
<td>A text description of the knowledge base.</td>
</tr>
<tr>
<td>Set default knowledge field values</td>
<td>Default values for knowledge articles in this publication. To define a default value, select a field in the left column, then use the right column to enter the data to automatically populate the selected field. Knowledge contributors can choose to apply default field values when selecting a knowledge base for an article.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot set a default value for the <strong>Author</strong> field.</td>
</tr>
</tbody>
</table>

**Knowledge workflows**

The publishing and retirement processes for a knowledge article are controlled by workflows defined for the knowledge base that the article belongs to.

You can assign different workflows to each knowledge base.

You can use one of the default workflows, or create your own workflows to define custom publishing and retirement processes for different types of knowledge.

For the workflows that require approval, you can configure which users can approve or reject by editing the `getApprovers()` function in the `KBWorkflow` script include.
# Default knowledge workflows

<table>
<thead>
<tr>
<th>Workflow</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge - Approval Publish</td>
<td>Requests approval from a manager of the knowledge base before moving the article to the published state. The workflow is canceled and the article remains in the draft state if any manager rejects the request.</td>
</tr>
<tr>
<td>Knowledge - Approval Retire</td>
<td>Requests approval from a manager of the knowledge base before moving the article to the retired state. The workflow is canceled and the article remains in the published state if any manager rejects the request.</td>
</tr>
<tr>
<td>Knowledge - Instant Publish</td>
<td>Immediately publishes a draft article without requiring an approval.</td>
</tr>
<tr>
<td>Knowledge - Instant Retire</td>
<td>Immediately retires a published article without requiring an approval.</td>
</tr>
<tr>
<td>Knowledge - Publish Knowledge</td>
<td>A subflow that moves the knowledge article to the published state. You can use this subflow when defining your own workflow.</td>
</tr>
<tr>
<td>Knowledge - Retire Knowledge</td>
<td>A subflow that moves the knowledge article to the retired state. You can use this subflow when defining your own workflow.</td>
</tr>
</tbody>
</table>

## Knowledge properties

As an administrator, you can configure the look and functionality of many knowledge base features with knowledge properties.

You can access knowledge properties by navigating to **Knowledge > Properties**.

## Knowledge portal properties

Knowledge portal properties control the look and functionality of the legacy portal page. This portal page lists recent knowledge articles and their publishing dates, organized by topic.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of columns on Knowledge portal pages.</td>
<td>Set the number of columns for arranging topics on the knowledge portal.</td>
</tr>
</tbody>
</table>

**Note:** This property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of articles per content block on the home page (glide.knowman.content_block_limit)</td>
<td>Enter the maximum number of articles to display in the <strong>Most Useful</strong> and <strong>Most Viewed</strong> sections of the knowledge homepage.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages.</td>
</tr>
<tr>
<td>Omit empty Knowledge topics and categories.</td>
<td>Select the Yes check box to display topic sections only if there are articles for the topic. Clear the check box to show all topic sections, even those with no assigned articles.</td>
</tr>
<tr>
<td>(glide.knowman.show_only_populated)</td>
<td><strong>Note:</strong> This property does not apply to the <strong>News</strong> category.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This property applies only to the default knowledge base and the legacy knowledge portal. Knowledge v3 articles do not specify a topic. Therefore, this property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages.</td>
</tr>
<tr>
<td>Show Knowledge section descriptions.</td>
<td>Select the Yes check box to display topic descriptions in the knowledge portal. To enter these descriptions:</td>
</tr>
<tr>
<td>(glide.knowman.show_descriptions)</td>
<td>1. Edit an existing knowledge article.</td>
</tr>
<tr>
<td></td>
<td>2. Right-click the <strong>Topic</strong> field label and choose <strong>Show Choice List</strong>.</td>
</tr>
<tr>
<td></td>
<td>3. Open the record for the topic name and enter the description into the <strong>Hint</strong> field.</td>
</tr>
<tr>
<td></td>
<td>Clear the check box to omit section descriptions from the knowledge portal.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages.</td>
</tr>
<tr>
<td>Number of Knowledge Base items to preview in a section.</td>
<td>Set the maximum number of articles per topic for the knowledge portal. The specific articles shown depends on the <strong>Knowledge section sort field</strong> property setting.</td>
</tr>
<tr>
<td>(glide.knowman.section_limit)</td>
<td><strong>Note:</strong> This property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| **Knowledge section sort field.** *(glide.knowman.section_sort)* | Select the default order for articles within each topic section on the knowledge portal.  
- **Published:** uses the date published.  
- **View count:** uses the number of times articles have been viewed during the period specified in the **Number of days used when summing article views** property.  
- **Alphabetically:** uses the first letter of the article title *(Short description field)*.  

*Note:* This property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages. |
| **Knowledge section sort direction.** *(glide.knowman.section_sort_direction)* | Select the order, **ascending** or **descending**, in which articles are listed within each topic section on the knowledge portal. The value used for sorting is set in the **Knowledge section sort field** property.  

*Note:* This property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages. |
<p>| <strong>Show unpublished articles in Knowledge Base portal and topic lists.</strong> <em>(glide.knowman.show_unpublished)</em> | Select the <strong>Yes</strong> check box to allow users to see unpublished articles in the knowledge portal and knowledge search results. Use the subsequent <strong>List of roles...</strong> properties to designated which users can see articles in various unpublished states. If this is not selected, users with appropriate roles can access unpublished articles through other modules of the <strong>Knowledge Base</strong> application. |
| <strong>List of roles (comma separated) that can see articles in the Review workflow state in the Knowledge portal and Topic list.</strong> <em>(glide.knowman.section.view_roles.review)</em> | Enter role names exactly as they appear in <strong>User Administration &gt; Roles</strong>. If <strong>Show unpublished articles in Knowledge Base portal and topic lists</strong> is selected, users with these roles see articles in the Review workflow state in the knowledge portal, in the topic list that appears when they click a topic title on the portal, and in knowledge search results. |
| <strong>List of roles (comma separated) that can see articles in the Draft workflow state in the Knowledge portal and Topic list.</strong> <em>(glide.knowman.section.view_roles.draft)</em> | Enter role names exactly as they appear in <strong>User Administration &gt; Roles</strong>. If <strong>Show unpublished articles in Knowledge Base portal and topic lists</strong> is selected, users with these roles see articles in the Draft workflow state in the knowledge portal, in the topic list that appears when they click a topic title on the portal, and in knowledge search results. |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define roles that can view articles in other/custon workflow states. Do not include Draft and Review states, as they are already defined in other properties. (Format - stage:[roles];) For example:</td>
<td>Enter role names exactly as they appear in User Administration &gt; Roles. If Show unpublished articles in Knowledge Base portal and topic lists is selected, define the custom field and role as workflow state:[roles];. The users with these roles see articles in the custom workflow state on the knowledge portal.</td>
</tr>
<tr>
<td>The number of articles that are asynchronously loaded when scrolling down in the new search results page. (glide.knowman.search.articles_per_page)</td>
<td>Enter the number of articles to be displayed during a search.</td>
</tr>
<tr>
<td>Show only Knowledge topic titles on portal page (no article links). (glide.knowman.home_titlesonly)</td>
<td>Select the Yes check box to omit all article titles from the knowledge portal and show only the topic names. Users can click the topic title to see the list of related articles.</td>
</tr>
<tr>
<td>Number of days (integer, default 30) used when summing article views. Views older than this are not considered when sorting articles based on view count. 0 means consider all views. (glide.knowman.view_age.days)</td>
<td>Enter a number of days to consider when calculating view count. This is used only when the Knowledge section sort field property is set to View count. Enter 0 to have ServiceNowServiceNow consider all views, regardless of date.</td>
</tr>
<tr>
<td>Automatically place cursor in Knowledge portal search box. (glide.knowman.portal_search_focus)</td>
<td>Select the Yes check box to have ServiceNowServiceNow place the cursor in the search field when a user opens the knowledge portal. Clear the check box to have users click in the search field before entering a search term.</td>
</tr>
</tbody>
</table>

**Knowledge search properties**

Knowledge search properties give you control over the search options available to users and the presentation of search results.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Knowledge search result order.  
( glide.knowman.order.search ) | Select the order for displaying search results:  
- **Number of Views**: orders search results based on the **Number of days used when summing article views** property.  
- **Relevancy**: orders search results based on content that is similar to the search term.  
- **Last Modified**: orders search results based on modification date. |
| Search method used when searching Knowledge from a task or directly in the Knowledge Base.  
( glide.knowman.search.operator ) | Select the search approach to use when the search term includes multiple words.  
- **OR query**: returns articles that contain at least one word from the search term.  
- **AND then OR query**: first searches for articles that include all words in the search term (an AND query). If no matches are found, an OR query is performed. |
| Show author in knowledge search results.  
( glide.knowman.search.show_author ) | Select this check box to include the author of each article in knowledge search results. |
| Show last modified date and time in knowledge search results.  
( glide.knowman.search.show_last_modified ) | Select this check box to include the date and time each article was last edited in knowledge search results. |
| Show publish date in knowledge search results.  
( glide.knowman.search.show_published ) | Select this check box to include the date each article was published in knowledge search results. |
| Show category in knowledge search results.  
( glide.knowman.search.show_category ) | Select this check box to include the category breadcrumbs of each article in knowledge search results. |
| Show number of views in knowledge search results.  
( glide.knowman.search.show_view_count ) | Select this check box to include the number of times each article was viewed in knowledge search results. |
| How to display attachments in Knowledge Search Results.  
( glide.knowman.search.attachment ) | Select the behavior of how the attachments are shown in the search results:  
- **Do not show attachment**  
- **Show only link for attachment**  
- **Show attachment with text snippet** (default)  
  For best performance, select **Do not show attachment**, or **Show only link for attachment**. |
| Show relevancy in knowledge search results.  
( glide.knowman.search.show_relevancy ) | Select this check box to show how relevant each search result is based on the search string.  
Relevancy value is only shown when Knowledge search result order property is set to Relevancy. |
### Property view properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable instant search results for knowledge. (glide.knowman.search.instant_results)</td>
<td>Select this check box to enable instant search for search results. The default is No.</td>
</tr>
</tbody>
</table>
| Show article number in knowledge search results. (glide.knowman.search.show_article_number) | Select this check box to display the article number for each article in the search result.  
**Note:** This property does not apply to contextual search results. |

### Knowledge article view properties

Article view properties give you control over the meta data and rating options that appear with articles. They also enable you to grant access to individual feedback options based on user role.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show tasks to which an article has been recently attached. (glide.knowman.recent_tasks.display)</td>
<td>Select the <strong>Yes</strong> check box to have article view include a list of tasks associated with the article. Knowledge articles can be attached to a task by clicking the <strong>Attach to Task</strong> button after searching from a form.</td>
</tr>
<tr>
<td>Number of attached tasks to display when viewing an article. (glide.knowman.recent_tasks)</td>
<td>Specify the maximum number of tasks to list in article view.</td>
</tr>
</tbody>
</table>
| Show article rating section, which may optionally include yes/no rating, star rating, and flagging options. (glide.knowman.show_rating_options) | Select the **Yes** check box to display the rating options for users with specific roles. Clear this check box to omit the rating section for all users, regardless of role.  
**The Feedback** field is always displayed for all users. |
| List of roles (comma separated) that can see an article's rating section, which may optionally include yes/no rating, star rating, and flagging option. (glide.knowman.show_rating_options.roles) | Enter role names exactly as they appear in **User Administration > Roles**. If **Show article rating section...** is selected, users with the roles listed here see the yes/no rating, star rating, and flagging options, according to related property settings.  
Be sure to include all roles that should be permitted to see any of these rating options. Separate properties enable you to identify specific roles for each option separately. All roles in those properties must also be listed here. |
<p>| Show the &quot;Was this article helpful?&quot; yes/no rating option. (glide.knowman.show_yn_rating) | Select the <strong>Yes</strong> check box to display the &quot;Was this article helpful?&quot; rating to users with roles set in the associated property. This appears only if the <strong>Show article rating section...</strong> property is selected. |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of roles (comma separated) that can see yes/no &quot;Was this article helpful?&quot; rating option.</td>
<td>Enter role names exactly as they appear in User Administration &gt; Roles. If both Show article rating section... and Show the &quot;Was this article helpful?&quot; properties are selected, users with the roles listed here see the “Was this article helpful?” rating in article view. All roles listed here must also be listed in the List of roles that can see an article’s rating section...&quot;property.</td>
</tr>
<tr>
<td>Show &quot;Create Incident&quot; link after a Knowledge article is rated not helpful.</td>
<td>Select the Yes check box to display the Create Incident link after a user rates an article as not helpful.</td>
</tr>
<tr>
<td>URL used for the &quot;Create Incident&quot; link after rating a Knowledge article not helpful.</td>
<td>Enter the URL for the page where users can create an incident only if the Show &quot;Create Incident&quot; link...&quot; property is selected.</td>
</tr>
<tr>
<td></td>
<td>To find the URL, open the appropriate page, then right-click the header bar and choose Copy URL. Highlight the URL that appears and use the browser’s copy and paste feature to place it into this property field.</td>
</tr>
<tr>
<td>Show five-star rating option.</td>
<td>Select the Yes check box to display the five-star rating to users with roles set in the associated property. This appears only if the Show article rating section... property is selected.</td>
</tr>
<tr>
<td>List of roles (comma separated) that can see five-star rating option.</td>
<td>Enter role names exactly as they appear in User Administration &gt; Roles. If both Show article rating section... and Show five-star rating option properties are selected, users with the roles listed here see the five-star rating option in article view. All roles listed here must also be listed in the List of roles that can see an article’s rating section...&quot;property.</td>
</tr>
<tr>
<td>List of roles (comma separated) that can flag incomplete/inaccurate articles.</td>
<td>Enter role names exactly as they appear in User Administration &gt; Roles. If both Show article rating section... and Show &quot;Flag Article&quot; option properties are selected, users with the roles listed here see the flag article option in article view. All roles listed here must also be listed in the List of roles that can see an article’s rating section...&quot;property.</td>
</tr>
<tr>
<td>Show user comments on knowledge articles.</td>
<td>Select an option for showing user comments When article loads, When user clicks link to show comments, or Never for users with roles set in the associated property:</td>
</tr>
</tbody>
</table>
### Knowledge homepage properties

Knowledge homepage properties control the featured content on the knowledge homepage.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default keyword for getting pinned articles.</td>
<td>Enter a default keyword for pinned articles. Articles pinned with the specified keyword appear in the Featured Content section of the knowledge homepage.</td>
</tr>
<tr>
<td>Default header title for the pinned articles section on knowledge home pages.</td>
<td>Enter a title for the Featured Content section of the knowledge homepage.</td>
</tr>
<tr>
<td>Display or hide the count of articles and questions in the category and child categories</td>
<td>Select to display the count of articles and questions within each category, including subcategories, when browsing or searching knowledge bases.</td>
</tr>
</tbody>
</table>

### Other knowledge properties

Additional knowledge properties let you control general knowledge management features.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When attaching an article to an incident, copy the content into this field.</td>
<td>When a user searches knowledge from a task form (such as an incident, problem, or change) and clicks the <strong>Attach to Task</strong> button for an article, the system copies the article number and content into the field specified here. You can specify a separate field for each table you want to attach articles to as a comma-separated list. Use dot walking to copy the article content into related records. If a specified field does not exist, that field is ignored.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Knowledge Management logo to display if running out of the ServiceNow frames. (glide.knowman.frameless_logo)</td>
<td>Click the reference lookup icon (🔍) and choose an image file to be displayed on knowledge pages that are used outside of the ServiceNow application frames. For example, the logo image selected here appears if you create a Content Management System site that includes knowledge pages. You can also enter the Name of an image stored in the database to use that image. <strong>Note:</strong> This property applies only to Knowledge v2 pages. This property is not supported on Knowledge v3 pages.</td>
</tr>
<tr>
<td>Hide the 'Import' functionality (button and drag-n-drop) for all users. (glide.knowman.import.hide_import_functionality)</td>
<td>Select Yes to hide the Import Articles button. If you have not yet migrated to Knowledge v3 and are using legacy Knowledge v2, you can configure the visibility of the Import Articles button to prevent users from importing articles to knowledge bases.</td>
</tr>
<tr>
<td>Show publish check box on the knowledge import pop-up. (glide.knowman.import.show_publish_checkbox)</td>
<td>Select No to remove the Publish check box on the knowledge import form so that an article cannot be published as part of the import process. Default is yes.</td>
</tr>
<tr>
<td>Use submission workflow. (glide.knowman.submission.workflow)</td>
<td>Select the Yes check box to use the knowledge submission workflow instead of the standard knowledge workflow. If this option is selected, each time a user creates knowledge from an incident or problem, the content is placed into a submission record instead of a draft article. Transforming knowledge submissions into articles follows a specific workflow.</td>
</tr>
<tr>
<td>The sys_id of the knowledge base when creating knowledge from task records. (glide.knowman.task_kb)</td>
<td>Enter the sys_id of the knowledge base to put new knowledge articles created from tasks in.</td>
</tr>
<tr>
<td>Before displaying an article's text in search results or the article view, check field-level ACLs on appropriate field: kb_knowledge.text (HTML article) or kb_knowledge.wiki (wiki text article). (glide.knowman.text.check_can_read)</td>
<td>Select this check box if users must have ACL-based permission to view text fields on the Knowledge [kb_knowledge] table to read article previews in knowledge search results.</td>
</tr>
</tbody>
</table>

Create a custom knowledge homepage

As a system administrator, you can create a module allowing users to open a knowledge homepage for a specific knowledge base or category.
Role required: admin

1. Right-click the application label in the application navigator and select **Edit Application**.
2. In the **Modules** related list click **New**.
3. Enter a **Title** for the module.
4. In the **Link type** field, select **URL (from Arguments)**.
5. In the **Arguments** field, append one or more of the following values to `$knowledge.do#/search?`.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_kb=&lt;knowledge base sys_id&gt;</td>
<td>Enter the sys_id of a knowledge base to show knowledge articles from that knowledge base by default. If the specified knowledge base does not exist, the default knowledge homepage appears when accessing the custom homepage module.</td>
</tr>
<tr>
<td>sysparm_category=&lt;knowledge category sys_id&gt;</td>
<td>Enter the sys_id of a knowledge category to show knowledge articles from that category by default. If the specified category does not exist, the default category for the selected knowledge base appears when accessing the custom homepage module. Note: The category passed in this parameter must belong to the knowledge base passed in the sysparm_kb parameter.</td>
</tr>
<tr>
<td>sysparm_order=&lt;view_count, last_modified, or relevancy&gt;</td>
<td>Enter the default sort order for articles to appear in.</td>
</tr>
</tbody>
</table>

To create a homepage for a **Facilities** knowledge base with a **sys_id** of d582764047022100158b949b6c9a7145 and sorting by last modified date, enter `$knowledge.do#/search?sysparm_kb=d582764047022100158b949b6c9a7145&sysparm_order=last_modified` in the **Arguments** field on the Module form.

**I18N - Knowledge internationalization**

Organizations with knowledge users who speak multiple languages can activate the optional knowledge internationalization features.

Activating internationalization plugins for any of the available languages automatically activates the I18N: Knowledge Management Internationalization Plugin v2 plugin (com.glideapp.knowledge.i18n2).

When active, knowledge internationalization enables the knowledge management team to create language-specific knowledge articles and keep translations of the same article related to each other so they are easy to manage. Users can view and search within their own language while still being able to view articles in other languages when necessary.
**Note:** To set the default language for searching articles to a language different from the logged in language of the user, specify the desired language in the `glide.knowman.search.default_language` system property.

For example, if you are logged in using the French language and would like the default language for searched articles to be English, specify English in the system property. You can then switch to French from within the UI to see French articles.

If no language is specified in the system property, articles default to the logged in language of the user.

Social Q&A does not use language-specific questions and answers. All questions and answers appear in the language they were created in. For example, a question in English is not translated when the current user's selected language is French.

**Activate the knowledge management internationalization plugin**

To enable translation of knowledge articles, you must activate multiple plugins.

- **I18N:Knowledge Management Internationalization v2 plugin**
- Internationalization plugins for each appropriate language (for example, I18N: French Translations and I18N: Korean Translations)

*Activate a plugin* on page 1233

**Installed Components**

Activating the Knowledge Management Internationalization plugin installs these components:

- A relationship named Translated Versions that creates a related list on the Knowledge form showing other articles that have the same parent article.
- A business rule named knowledge query that automatically filters the knowledge portal and search results based on the user's selected language.
- A UI macro named kb languages.
- Two fields, named Language and Parent, in the Knowledge table and the Knowledge form.

**Knowledge Form Changes**

Activating knowledge management internationalization adds these fields to the Knowledge form.

- **Language:** select the language for the article.
- **Parent:** enter the number of the article that represents the base language. This setting keeps translations of the same article related to each other. Consider choosing the same language consistently for the parent article.

Additionally, the Translated Versions related list is automatically added to the Knowledge form after you save an article.

**Create a translation for a knowledge article**

Create a translation for an article from the Translated Versions related list on the knowledge article record. Before creating a translation, activate the internationalization plugin for the desired
language which, in turn, activates the I18N: Knowledge Management Internationalization Plugin v2 (com.glideapp.knowledge.i18n2).

1. Open the knowledge article record.
2. In the Translated Versions related list, click New.
3. Select the desired language for the translation in the Language field.
4. Provide a description in the Short Description field.
5. Click Submit.

The translation is created and added to the Translated Versions related list.

Use knowledge on mobile devices

All ServiceNow ITSA Suite users can access knowledge from mobile devices to search for and view knowledge articles and social Q&A questions.

Browse knowledge on mobile devices

You can browse knowledge bases using a mobile device.

Navigate to SELF-SERVICE > Knowledge Base.
Browse Knowledge by tapping on a category and browsing articles or subcategories.

View knowledge articles and questions on mobile devices
You can read knowledge content using a mobile device.
Tap a title to view the article or question.

**Note:** You cannot edit or add comments to knowledge articles on the mobile interface. If using Social Q&A on page 2132, you can perform all functions such as adding and editing questions, answers, and comments through the mobile interface.
Search knowledge on mobile devices

Find a specific article or question by searching on a particular term.
Knowledge Management v3 migration

Knowledge management is significantly changed with knowledge v3, which is enabled by default for all instances starting with the Fuji release.

Knowledge v3 has several key differences from the version of knowledge management available with the Eureka release, knowledge v2.

When upgrading from Eureka or earlier, you must migrate legacy knowledge content and any customizations you have made in the Knowledge Base application in order to use the full range of new functionality.

This video demonstrates how to migrate from Knowledge Management v2 to v3.

Table 518: Key differences

<table>
<thead>
<tr>
<th>Legacy Knowledge</th>
<th>Knowledge v3</th>
</tr>
</thead>
<tbody>
<tr>
<td>One knowledge base.</td>
<td>Multiple knowledge bases.</td>
</tr>
<tr>
<td>A single lifecycle workflow shared by all articles.</td>
<td>Separate, customizable workflows available for each knowledge base.</td>
</tr>
<tr>
<td>Legacy Knowledge</td>
<td>Knowledge v3</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Two-level organizational structure with <strong>Topic</strong> and <strong>Category</strong>. A single</td>
<td>Category structure with any number of levels. Each knowledge base has a</td>
</tr>
<tr>
<td>organizational taxonomy shared by all articles.</td>
<td>unique category taxonomy.</td>
</tr>
<tr>
<td>Permissions defined per article using roles and ACLs.</td>
<td>Permissions defined per knowledge base using user criteria.</td>
</tr>
</tbody>
</table>

### Knowledge Management v3 migration process

When you upgrade from Eureka or earlier, consider a multi-step migration process. This process ensures you can access existing knowledge content and allows you to fully migrate at your own pace.

Review automatic changes before upgrading. These changes are applied immediately when upgrading and may impact existing functionality.

1. Migrate knowledge functionality to use your customized functionality with the knowledge v3 enhancements.
2. Migrate knowledge content to use the knowledge v3 enhancements with your existing knowledge articles.

### After migration to Knowledge v3

These items are relevant to getting the full range of new functionality after you migrate.

- If you have not upgraded to Geneva Patch 3, you must create a category structure in the destination knowledge bases for existing articles. Since knowledge v3 supports multiple knowledge bases, **Topic** and **Category** fields are deprecated and searching is only supported on the new **Category** and **Subcategory** fields. For usability and display purposes, we recommend limiting your category levels to five deep (or fewer).
- If you have not upgraded to Geneva Patch 3 and the Knowledge article form was customized in knowledge v2, you must open the article form and add the Knowledge Base field to allow creation of new articles. The Knowledge Base field is mandatory in knowledge v3.
- You must configure knowledge access controls based on user criteria to ensure that users can access existing knowledge content. Since access control in knowledge v3 is intended to be based on user criteria alone, users may be unable to access v2 or v3 knowledge articles, or have restricted access, with preexisting ACLs.
- If the home page was customized in knowledge v2, custom layout and links are not saved in knowledge v3.

### Automatic changes after KMv3 migration

Upgrading to Knowledge v3 automatically changes knowledge functionality including the knowledge user interface and security model.

### Default knowledge base

When you upgrade from Eureka or earlier, a default knowledge base is created, titled Knowledge, that contains all knowledge articles from the previous version.

After the upgrade, you can add and edit articles in this knowledge base.
Articles in the default knowledge base continue to use legacy functionality, such as role-based access controls. Any customizations you made to knowledge functionality before the upgrade are preserved in the default knowledge base.

Knowledge search changes

By default, the legacy knowledge portal uses knowledge v3 search with limited filtering options. You can filter results only by knowledge base. Search results include only those articles the current user can view based on user criteria. ACLs do not restrict search results.

Knowledge security changes

Knowledge v3 manages access to articles with user criteria rather than roles and ACLs. Existing ACLs on the Knowledge [kb_knowledge] table are preserved, but with the following changes.

- A version of 3 indicates that a knowledge base uses the knowledge v3 functionality.
- A version of 2 indicates that a knowledge base uses legacy functionality.

New ACLs introduced by knowledge v3, such as field-level controls, apply to all knowledge bases.

When you upgrade from Eureka or earlier, any custom ACLs you implemented for the Knowledge table still apply to any knowledge records in the default knowledge base. Knowledge v3 updates ACLs that exist before upgrade so they only apply to Knowledge [kb_knowledge] records where this field value is 2.

**Note:** These changes affect the default knowledge base. Even if you are not using the new knowledge functionality, the legacy knowledge behavior may change if you have made customizations to ACLs on the Knowledge [kb_knowledge] table. Ensure you configure knowledge access controls after upgrading to prevent unexpected behavior.

Because Knowledge v3 introduces ACLs for the Knowledge [kb_knowledge] table, if you do not have high security enabled, you may notice differences in default behavior. See KB0549970 for information about configuring Knowledge v3 without high security.

Knowledge article view changes

The Knowledge application uses the kb_view UI page to display knowledge articles.

**UI page**

The following changes apply automatically when you upgrade from Eureka or earlier:

- The existing kb_view page is renamed to kb_view_customer. The kb_view_customer page is used to display articles in the default knowledge base, preserving the legacy user experience.
- A new kb_view page is used. This becomes the primary page used to display articles that are not in the default knowledge base.

You should review all customizations introduced around the legacy kb_view UI page.

- URLs should not be updated as the new kb_view page redirects to the correct page depending on the article.
- Customizations made to the kb_view UI page are automatically copied to the kb_view_customer page on upgrade.
- Other customizations that affect kb_view may need to be copied or modified to apply to kb_view_customer. For example, an ACL defined for kb_view may need to be duplicated to provide security coverage for kb_view_customer.
**Feedback**

Legacy feedback functionality is retained for knowledge articles displayed in the legacy article view. Users can continue to use the feedback options to view, add, and reply to comments on these legacy articles. Legacy comments are retained in the Knowledge Feedback [kb_feedback] table.

To retain the legacy comment functionality with new knowledge articles, set the Use Live Feed for Knowledge feedback property (glide.knowman.use_live_feed) to false. This property is true by default.

**Knowledge portal changes**

The legacy knowledge portal, kb_home, is preserved in knowledge v3.

**UI page**

You can search articles from the knowledge portal and filter search results by knowledge base, and by language if the instance uses knowledge management internationalization. This portal also appears for users on older browsers, such as Internet Explorer 9, instead of the knowledge homepage.

Navigation add-ons are available in the legacy knowledge portal. Navigation add-ons of the Link type appear in the top-right of the portal. To add navigation add-ons of the Search type, you must customize the kb_home UI page.

**Search**

By default, the legacy knowledge portal uses knowledge v3 search with limited filtering options. You can filter results only by knowledge base. Search results include only those articles the current user can view.

**Knowledge submissions**

Knowledge submissions created from tasks continue to work in knowledge v3 with minor changes.

By default, the **Topic** and **Category** fields no longer appear on the Submission form due to the knowledge v3 category changes.

The property glide.knowman.submission.workflow no longer appears on the knowledge properties UI page.

To enable knowledge submissions, set this property to true on the System Properties [sys_properties] table.

See creating knowledge from incidents and problems for instructions on using knowledge submission in knowledge v3.

**Migrating Knowledge access controls**

Access to knowledge articles is controlled based on the user criteria for the specific knowledge base.

**Important:** If you do not migrate existing knowledge ACLs to user criteria, users may be unable to access knowledge articles, or may be able to access restricted articles.

User criteria records, rather than ACLs, control basic create, read, update, and delete operations for knowledge articles as well as which search results appear. This may lead to unexpected behavior if your configuration depends on ACLs to control access.

To ensure users are able to access the correct knowledge articles, migrate any ACL-based security settings to user criteria. This process may include creating additional knowledge bases and moving
existing content to these knowledge bases, depending on your security model. For a detailed explanation
and migration examples, refer to KB0550924 Understanding User Criteria and ACLs in Knowledge v3.

This video demonstrates how to manage article-level access controls during migration from Knowledge
Management v2 to v3.

**Knowledge functionality migration**

You can migrate knowledge functionality to use your customized functionality with the knowledge v3 enhancements.

**Replaced knowledge modules**

When you are ready to migrate to knowledge v3, you can activate the v3 modules and deactivate the legacy modules.

Use the tables below to identify which knowledge v3 modules to activate and which legacy modules to deactivate. When activating a knowledge v3 module, deactivate the knowledge v2 module with the same name to ensure users access the correct module.

You can use the enablev3anddisablev2menus.txt backup script to enable knowledge v3 menus and disable knowledge v2 menus.

**Table 519: Knowledge v3 modules**

<table>
<thead>
<tr>
<th>Title</th>
<th>Order</th>
<th>Link type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homepage</td>
<td>100</td>
<td>URL (from Arguments:)</td>
</tr>
<tr>
<td>Articles</td>
<td>200</td>
<td>Separator</td>
</tr>
<tr>
<td>Create New</td>
<td>300</td>
<td>New Record</td>
</tr>
<tr>
<td>Unpublished</td>
<td>400</td>
<td>List of Records</td>
</tr>
<tr>
<td>Published</td>
<td>500</td>
<td>List of Records</td>
</tr>
<tr>
<td>Retired</td>
<td>600</td>
<td>List of Records</td>
</tr>
<tr>
<td>Flagged</td>
<td>800</td>
<td>List of Records</td>
</tr>
<tr>
<td>All</td>
<td>900</td>
<td>List of Records</td>
</tr>
<tr>
<td>Open Submissions</td>
<td>960</td>
<td>List of Records</td>
</tr>
<tr>
<td>Administration</td>
<td>1000</td>
<td>Separator</td>
</tr>
<tr>
<td>Knowledge Bases</td>
<td>1100</td>
<td>List of Records</td>
</tr>
<tr>
<td>Feedback</td>
<td>1300</td>
<td>List of Records</td>
</tr>
<tr>
<td>Ratings</td>
<td>1500</td>
<td>List of Records</td>
</tr>
<tr>
<td>Search Log</td>
<td>1600</td>
<td>List of Records</td>
</tr>
<tr>
<td>Navigation Add-ons</td>
<td>1700</td>
<td>List of Records</td>
</tr>
<tr>
<td>Messages</td>
<td>1800</td>
<td>List of Records</td>
</tr>
<tr>
<td>Properties</td>
<td>1900</td>
<td>URL (from Arguments:)</td>
</tr>
<tr>
<td>Overview</td>
<td>2000</td>
<td>URL (from Arguments:)</td>
</tr>
<tr>
<td>User Criteria</td>
<td>2100</td>
<td>List of Records</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
### Table 520: Legacy modules

<table>
<thead>
<tr>
<th>Title</th>
<th>Order</th>
<th>Link type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create New</td>
<td>100</td>
<td>New Record</td>
</tr>
<tr>
<td>Published</td>
<td>190</td>
<td>List of Records</td>
</tr>
<tr>
<td>Edit</td>
<td>200</td>
<td>List of Records</td>
</tr>
<tr>
<td>Retired</td>
<td>240</td>
<td>List of Records</td>
</tr>
<tr>
<td>Feedback</td>
<td>300</td>
<td>List of Records</td>
</tr>
<tr>
<td>View</td>
<td>400</td>
<td>URL (from Arguments:)</td>
</tr>
<tr>
<td>Submissions</td>
<td>410</td>
<td>Separator</td>
</tr>
<tr>
<td>Assigned to me</td>
<td>420</td>
<td>List of Records</td>
</tr>
<tr>
<td>Open Submissions</td>
<td>450</td>
<td>List of Records</td>
</tr>
<tr>
<td>KCS</td>
<td>460</td>
<td>List of Records</td>
</tr>
<tr>
<td>Flagged Articles</td>
<td>470</td>
<td>List of Records</td>
</tr>
<tr>
<td>Ratings</td>
<td>480</td>
<td>List of Records</td>
</tr>
<tr>
<td>Searching Log</td>
<td>495</td>
<td>List of Records</td>
</tr>
<tr>
<td>Overview</td>
<td>496</td>
<td>URL (from Arguments:)</td>
</tr>
<tr>
<td>Administration</td>
<td>500</td>
<td>Separator</td>
</tr>
<tr>
<td>Navigation Add-ons</td>
<td>600</td>
<td>List of Records</td>
</tr>
<tr>
<td>Properties</td>
<td>700</td>
<td>URL (from Arguments:)</td>
</tr>
<tr>
<td>Messages</td>
<td>800</td>
<td>List of Records</td>
</tr>
</tbody>
</table>

#### Knowledge article publish and retire workflows

With knowledge v3, article state is controlled by workflows.

In the legacy knowledge base, article state was controlled by UI actions. UI actions from previous versions are available only in the default knowledge base.

To preserve your article publishing and retirement process in new knowledge bases, create workflows that follow these processes. To create these workflows, copy the default workflow that best matches your process and modify that copy. After creating workflows that use your article publishing and retirement processes, use these workflows as the **Publish workflow** and **Retire workflow** for the new knowledge bases.

#### Knowledge article category structure

With knowledge v3, all knowledge articles are organized by category and subcategory within knowledge bases.

The default knowledge base uses the legacy category structure; knowledge articles are organized by topics and categories.

Before you migrate articles, create a category structure for these articles in the destination knowledge bases. This structure can reproduce your legacy topic and category structure or can be extended to take advantage of the knowledge v3 category functionality.
Knowledge content migration

To use the new functionality available in knowledge v3 with legacy articles, you must migrate these articles out of the default knowledge base.

To migrate knowledge content, move articles out of the default knowledge base by changing the Knowledge Base value for each article. The state of the article remains unchanged when moving articles. After selecting a new knowledge base, assign a category to each article using the category structure defined in the new knowledge base.

After you migrate a knowledge article, the legacy feedback functionality is automatically replaced with the new knowledge feedback mechanisms. Legacy feedback is copied to the Live Feed Message [live_message] table to ensure previously submitted feedback is available for knowledge v3 articles.

Dependency Views

Dependency Views graphically displays an infrastructure view for a configuration item and the business services that it is part of and that it supports. Dependency Views indicates the status of its configuration items, and allows access to the CI's related alerts, incidents, problems, changes, and business services.

What's new

- Dependency Views release notes

Administer

- Create or modify Dependency Views map indicators on page 2189
- Create or modify Dependency Views map icons on page 2193
- Create or modify Map Related Items on page 2191
- Create or modify Dependency Views menu actions on page 2193

Use

- Dependency Views map on page 2180
- Dependency Views map menus and controls on page 2181
- Access Dependency Views map on page 2185
- Change the layout of Dependency Views map on page 2186
- Filter the view of a Dependency Views map on page 2186
- Perform actions on nodes in a Dependency Views map on page 2187

Develop

- Properties installed with Dependency Views on page 2176
- Tables installed with Dependency Views on page 2176

Troubleshoot and get help

- Supported browsers for Dependency Views on page 2184
- Search the HI knowledge base for known error articles
- Contact ServiceNow Support
- ServiceNow Community

Installed with Dependency Views

Dependency Views plugin adds tables and system properties.
Tables installed with Dependency Views

Dependency Views adds the following tables.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available CI icons</td>
<td>Stores all available CI class icons.</td>
</tr>
<tr>
<td>[ngbsm_ci_icons]</td>
<td></td>
</tr>
<tr>
<td>Icons for CI types</td>
<td>Maps icons to CI class names.</td>
</tr>
<tr>
<td>[ngbsm_ci_type_icon]</td>
<td></td>
</tr>
<tr>
<td>Map View</td>
<td>Serialized map views saved by users.</td>
</tr>
<tr>
<td>[ngbsm_view]</td>
<td></td>
</tr>
<tr>
<td>Map Filter</td>
<td>Filters saved by users.</td>
</tr>
<tr>
<td>[ngbsm_filter]</td>
<td></td>
</tr>
<tr>
<td>Menu Action</td>
<td>Default and custom context menu actions that appear when users right click a map.</td>
</tr>
<tr>
<td>[ngbsm_context_menu]</td>
<td></td>
</tr>
<tr>
<td>Related Item</td>
<td>Stores which reference fields should be treated as relationships when building the map. This allows users to include CI's that are related via a reference field instead of a relationship.</td>
</tr>
<tr>
<td>[ngbsm_related_item]</td>
<td></td>
</tr>
<tr>
<td>Edge Colors</td>
<td>Color definitions to use when drawing the relationships between nodes based on relationship type.</td>
</tr>
<tr>
<td>[bsm_edge_color]</td>
<td></td>
</tr>
<tr>
<td>Map Indicator</td>
<td>Stores all map indicators.</td>
</tr>
<tr>
<td>[bsm_indicator]</td>
<td></td>
</tr>
<tr>
<td>BSM Saved Map</td>
<td>Details of maps.</td>
</tr>
<tr>
<td>[bsm_graph]</td>
<td></td>
</tr>
<tr>
<td>BSM Map Actions</td>
<td>Actions on the map.</td>
</tr>
<tr>
<td>[bsm_action]</td>
<td></td>
</tr>
</tbody>
</table>

Properties installed with Dependency Views

Dependency Views adds the following properties.

**Note:** To open the System Property [sys_properties] table, enter `sys_properties.list` in the navigation filter.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.bsm.max_nodes                            | Maximum number of CI’s to display on a map at once  
Maximum number of CI’s to display on a map at once. The maximum number of nodes to retrieve from the database for a CI. If more nodes exist in the database, they are not displayed in the map.  
- **Type**: Integer  
- **Default value**: 1000  
- **Location**: Dependency Views > Map Properties |
| glide.bsm.too_many_children                    | Maximum number of child nodes to display (the rest will be collapsed)  
The maximum number of child nodes to display. Nodes are collapsed for the map to meet this limit.  
- **Type**: Integer, valid values 1 or greater  
- **Default value**: 10  
- **Location**: Dependency Views > Map Properties |
| glide.ngbsm.filters_remove_filtered_items      | Filtered out items should be removed from the graph along with any disjoint children  
Filtered out items should be removed from the graph along with any disjoint children.  
- **Type**: Yes | No  
- **Default value**: No  
- **Location**: Dependency Views > Map Properties |
| glide.ngbsm.filters_run_layout_automatically    | When filters are changed the graph will recalculate the layout using the currently selected layout algorithm  
When filters are changed the graph will recalculate the layout using the currently selected layout algorithm.  
- **Type**: Yes | No  
- **Default value**: No  
- **Other possible values**:  
  - Value 1: Yes  
- **Location**: Dependency Views > Map Properties |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.ngbsm.filters_fit_to_screen_automatically                           | When filters are changed the graph should be fit to the screen.  
When filters are changed the graph should be fit to the screen.  
• **Type**: Yes | No  
• **Default value**: No  
• **Other possible values**:  
  • **Value 1**: Yes  
• **Location**: Dependency Views > Map Properties                                                                                              |
| glide.ngbsm.performance_allow_curves                                     | Allow links between nodes to be drawn using smooth curves (May impact performance)  
Allow links between nodes to be drawn using smooth curves (May impact performance)  
• **Type**: Yes | No  
• **Default value**: Yes  
• **Other possible values**:  
  • **Value 1**: No  
• **Location**: Dependency Views > Map Properties                                                                                              |
| glide.ngbsm.notification_display_time                                    | Amount of time in milliseconds a notification stays on the screen  
Amount of time in milliseconds a notification stays on the screen in Dependency Views  
• **Type**: Integer  
• **Default value**: 5000  
• **Location**: Dependency Views > Map Properties                                                                                              |
| glide.ngbsm.search_ci_limit                                             | Maximum amount of results displayed when searching for CIs  
Maximum amount of results displayed when searching for CIs in Dependency Views  
• **Type**: Integer  
• **Default value**: 10  
• **Location**: Dependency Views > Map Properties                                                                                              |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ngbsm.search_service_limit</td>
<td>Maximum amount of results displayed when searching for Services</td>
</tr>
<tr>
<td></td>
<td>Maximum amount of results displayed when searching for Services in Dependency Views</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: 5</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: Dependency Views &gt; Map Properties</td>
</tr>
<tr>
<td>glide.ngbsm.search_rel_type_limit</td>
<td>Maximum amount of results displayed when searching for Relationship Types</td>
</tr>
<tr>
<td></td>
<td>Maximum amount of results displayed when searching for Relationship Types in Dependency Views</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: 5</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: Dependency Views &gt; Map Properties</td>
</tr>
<tr>
<td>glide.bsm.color.affect_neighbors</td>
<td>Color of an affected neighbor node’s label</td>
</tr>
<tr>
<td></td>
<td>Used for backwards compatibility support.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: Color</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: #feefb2 (Beige)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Other possible values</strong></td>
</tr>
<tr>
<td></td>
<td>• A <em>valid HTML4 color</em> name. The color names are case-insensitive.</td>
</tr>
<tr>
<td></td>
<td>• An RGB hex code, such as #003366</td>
</tr>
<tr>
<td></td>
<td>• An RGB decimal code, such as rgb(0,51,102)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: Dependency Views &gt; Map Properties</td>
</tr>
<tr>
<td>glide.bsm.refresh_interval</td>
<td>This property has no effect on the map, and remains for backward compatibility purposes only.</td>
</tr>
<tr>
<td>glide.ngbsm.link_to_eureka</td>
<td>This property has no effect, and remains for backward compatibility purposes only.</td>
</tr>
</tbody>
</table>
## Dependency Views map

Dependency Views maps graphically display configuration items that support business services and the relationships between the configuration items.

A ServiceNow business service is work or goods that are supported by an IT infrastructure. For example, delivering email service to an employee can require services such as email servers, web servers, and the work to configure the user’s account.

A Dependency Views map has one starting point, called the root CI or root node of the map. The root CI is surrounded by a darker frame that repaints itself with a pulsing effect drawing the attention to the root CI. The maps can show both upstream and downstream dependencies for the root CI. By default the Dependency Views map displays 3 levels, both upstream and downstream relationships. Administrators can configure the number of levels displayed. The map collapses and expands clusters to make them easier to view. By default, clusters are collapsed.

In a Dependency Views map, map indicators indicate if a CI has any active, pending issues. You can investigate the tasks that are connected to a CI to get more details. When you return to the map from another form, the system restores the last map viewed, using the default filter and layout settings. When you click the map icon (🔍) on a CI record or on a task record that identifies a CI, the map opens.

Many of the relationships in map are created through the discovery process. You can also create, define, and delete CI relationships in the map. You can display the map from different perspectives and open specific records that relate to configuration items. The system refreshes the map automatically to reflect changes to the CMDB.

The Dependency Views module is active in all instances, and includes demo data.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ngbsm.show_class_labels</td>
<td>When available, the map should display the class labels for each CI. When available, the map should display the class labels for each CI. • <strong>Type</strong>: Yes</td>
</tr>
</tbody>
</table>
When you click the map icon ( on a CI record or on a task records that identify a CI, a map opens. You can change the **Link to the Eureka BSM when navigating via CI Reference Field or Related Items** property to have the Legacy BSM map application open instead.

### Roles

Users with the itil and ecmdb_admin roles can view maps and perform all actions in the map. Actions include access to the map views and saved filters, both from the lists in the map and from the **Saved Filters** module.

### Dependency Views map menus and controls

Dependency Views maps contain the following menus and controls.

### Map options

The following options are available across the top of the map.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Menu" /></td>
<td>Menu to save, load and export views of the map.</td>
</tr>
<tr>
<td><code>&lt;Root CI&gt;</code></td>
<td>Next to the menu icon is the name of the current root node (CI) of the map.</td>
</tr>
<tr>
<td><img src="image" alt="Search for CI" /></td>
<td>Enter the name of a CI or a business service to load into the map. Alternatively, you can start typing to have the auto-complete feature present a list of CIs and businesses service that match your partial value.</td>
</tr>
<tr>
<td>Vertical</td>
<td>Display the map in vertical view.</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Horizontal</td>
<td>Display the map in horizontal view.</td>
</tr>
<tr>
<td>Radial</td>
<td>Display the map in radial view.</td>
</tr>
<tr>
<td>Force</td>
<td>Centers the elements around the parent CI, regardless of upstream or downstream relationships.</td>
</tr>
<tr>
<td>Group</td>
<td>Groups the elements according to their CI type.</td>
</tr>
<tr>
<td>Details</td>
<td>Displays related lists such as Problems, Changes and Related Business Services that are associated with the selected CI. If the Event Management plugin is active, then events and alerts are also displayed.</td>
</tr>
<tr>
<td>Filters</td>
<td>View and create filters for the map.</td>
</tr>
</tbody>
</table>

Map menu

The following options are available if you right-click the map background.

<table>
<thead>
<tr>
<th>Run Layout</th>
<th>Redraws the map with the current layout option.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit To Screen</td>
<td>Resizes the map to fit all the nodes in the map window.</td>
</tr>
<tr>
<td>Reset Filters</td>
<td>Performs the same action as the Filters &gt; Reset option.</td>
</tr>
</tbody>
</table>

Node menu

The following options are available if you right-click a node.

<p>| View Form     | Displays the CMDB record of the selected CI in a new tab of the browser. |</p>
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Map</td>
<td>Reloads the map using the selected CI as the new root node, with the currently defined layout setting. This option does not display on the root node.</td>
</tr>
<tr>
<td>View Related Tasks</td>
<td>Displays all tasks or outages associated with the selected CI, including incidents, problems, change requests, and follow-on tasks. This option is always available, even if there are no tasks associated with the CI. This option does not appear on collapsed nodes.</td>
</tr>
<tr>
<td>View Affected CIs</td>
<td>Shows a list of all tasks that have the CI listed as an Affected CI. This option is only visible when you access the map from the map icon in a task record's Configuration item field.</td>
</tr>
<tr>
<td>View Related Outages</td>
<td>Displays all outages involving the selected CI. This option only appears when there is an outage associated with the CI. This option does not appear on collapsed nodes.</td>
</tr>
<tr>
<td>Add Relationship</td>
<td>This option displays a dotted green line that you can drag to another CI to create a relationship link. A popup dialog allows you to define the relationship type.</td>
</tr>
<tr>
<td>Expand</td>
<td>Displays all CIs and components within a clustered node, a collapsed node, or logical groups (visual nodes that appear when glide.bsm.too_many_children is reached). This option appears only if the node is a collapsed or cluster node.</td>
</tr>
<tr>
<td></td>
<td>The number of additional icons to display is bound by the value of the glide.bsm.max_nodes property.</td>
</tr>
<tr>
<td>Collapse</td>
<td>Collapses all CIs and components within a cluster node or a collapsed node back to a single node. It does not collapse logical groups that were expanded. This option only appears if the node has been expanded using the Expand menu item.</td>
</tr>
<tr>
<td>Run Layout From Here</td>
<td>This option re-runs the chosen layout using the current node. Use this option to get a new or clearer view on the same map.</td>
</tr>
<tr>
<td>Load More</td>
<td>Starting at the selected icon, loads the next level of the map, past the setting of Max Levels. The number of additional icons to display is bound by the value of the glide.bsm.max_nodes property.</td>
</tr>
</tbody>
</table>
Relationship menu

The following options are available if you right-click a relationship link.

<table>
<thead>
<tr>
<th>View Relationship Form</th>
<th>Opens the CI Relationship form. You can modify the Parent, Type, and Child of the relationship from this form.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify Relationship</td>
<td>Searches for and selects a new relationship for this link.</td>
</tr>
<tr>
<td>Delete Relationship</td>
<td>Deletes a relationship. The relationship is deleted after prompting for confirmation.</td>
</tr>
</tbody>
</table>

Supported browsers for Dependency Views

The latest version or service pack of internet browsers are required to view and manipulate Dependency Views maps.

The Dependency Views module supports the latest version or service pack of the following browsers:

- Firefox version 20 or later (latest version recommended)
- Chrome version 25 or later (latest version recommended)
- Safari version 6 or later
- Microsoft Internet Explorer (IE), with these requirements and limitations:
  - Dependency Views requires IE version 9 or later.
  - If you are using Internet Explorer 7 or 8, use the Legacy BSM map with Adobe Flash Player 10.1 or later.
  - Exporting images from a Dependency Views map using Internet Explorer as the browser is supported only for Internet Explorer version 11, and for Geneva Patch 9 releases and later.
  - When you save a map view, Dependency Views will not make a thumbnail image. You may navigate to the saved map view using the version number.

Collapsed nodes in a Dependency Views map

Dependency Views can display clusters and child nodes alongside individual CI nodes.

Aspects of the map can differ when cluster nodes are on the map. For example the nodes can display in a collapsed format to avoid unnecessary clutter in large maps. By default, the Dependency Views collapses all node clusters in the view. Menu options for a clustered node include Collapse and Expand which allow you to control the density on the map.

A cluster is an organized set of computer CIs that work together as a single system. Each node in the cluster represents a CI, typically a server, that can have referenced hardware, such as disks and network adapters. Clusters are CIs in the Cluster [cmdb_ci_cluster] table and can have tasks assigned to them.

A collapsed node collects all the children of a CI that are in the same class into a single node when the number of children in that class exceeds a configurable threshold. Collapsed nodes contain CIs but are not CIs themselves and cannot have tasks assigned to them.
Annotation

Icons for clusters and collapsed nodes are noted by the string "Cluster" and by a unique cluster icon. The system searches through all the component nodes in a cluster CI or collapsed node looking for tasks, outages, and trouble, such as incidents, problems, or change requests. This search evaluates only the number of levels that are displayed in the diagram.

Use Dependency Views

Use the layout controls on a Dependency Views map to display elements in different configurations for easier management. Use the filter panel on the map to display fewer levels or to filter out elements you don't want to see, then save the filter for use later. Draw new relationships between elements or edit existing relationships.

Access Dependency Views map

When you access a Dependency Views map from one of the view options, the view is centered on the root CI, and displays the layout and number of levels defined in the map properties.

The maps generated by Dependency Views are based on D3 and Angular technology, providing a modern interactive graphical interface to visualize configuration items and their relationships.

The maps that Service Mapping provides are for business services, including comprehensive maps from the perspective of business services. For more information, see Service Mapping.

Administrators can configure the setting for the default layout of the map and number of levels displayed. When you access the map from a saved view, the map opens using the properties in the saved view, and not the default map properties.

1. Navigate to Dependency Views and open one of these modules:
   - View Map in New Tab: Opens the map in a new, full screen tab without the application navigator.
   - View Map: Opens the map in the content pane of the current tab.
   - Saved Views: Opens a view of a map that you previously saved.

2. Click a number in the Version column, and then click the map icon.

Save or load a Dependency Views map

In the View Map module, use the menu icon to save and load Dependency Views maps.

1. Navigate to Dependency Views > View Map.
2. Click the menu icon.
3. Select Save View, Load View, or Last View.

Delete a saved Dependency Views map view

Use the Saved Views module to delete a previously saved view.

1. Navigate to Dependency Views > Saved Views.
2. Use the checkbox in the first column of the table to select the map view that you wish to delete.
3. Select **Delete** from the **Actions on selected rows** drop-down menu.

**Change the layout of Dependency Views map**

You can select from different layout options for your Dependency Views map.

1. Navigate to **Dependency Views > View Map**.
2. Select one of the following layout options from the menu across the top of the view.
   - **Vertical**: displays the elements in a vertical tree pattern according to their upstream and downstream relationships. This is the default value for the initial display of the map.
   - **Horizontal**: displays the elements in a horizontal tree pattern according to their upstream and downstream relationships.
   - **Radial**: displays the elements in a radial pattern according to their upstream and downstream relationships.
   - **Force**: centers the elements around the parent CI, regardless of upstream or downstream relationships.
   - **Group**: groups the elements according to their CI type.
   - **Details**: displays related alerts, incidents, problems, and business services.

**Filter the view of a Dependency Views map**

You can filter a Dependency Views map to display specific types or categories of configuration items.

Use the filter panel to control which elements of the map are displayed and to save versions of a filter for later use.

1. Navigate to **Dependency Views > View Map**.
2. Click the **Filters** button in the map's header bar to open the filter panel. Filter groups and controls in the filter panel are contained within collapsible strips. Click a strip to expand or collapse it. Within each strip, you can select individual items.

<table>
<thead>
<tr>
<th>Filter panel strips and options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create New Filter</td>
<td>Configure the desired filter settings and enter a name for the filter in this text box. Click <strong>Save</strong> and the filter will be available from the <strong>Load Saved Filter</strong> option.</td>
</tr>
<tr>
<td>Load Saved Filter</td>
<td>Apply a previously saved filter to the current map.</td>
</tr>
<tr>
<td>Max Levels</td>
<td>Designate how many levels from the root CI display on the map.</td>
</tr>
<tr>
<td>Filter CIs by Depth</td>
<td>Designate which levels of CI display on the map.</td>
</tr>
<tr>
<td>Filter CIs by CI Type</td>
<td>Designate what CI types display in the map.</td>
</tr>
<tr>
<td>Filter CIs By Location</td>
<td>Designate what CI locations display in the map.</td>
</tr>
<tr>
<td>Filter CIs By CI Manufacturer</td>
<td>Designate what CI manufacturers display in the map.</td>
</tr>
</tbody>
</table>
Perform actions on nodes in a Dependency Views map

You can view various related items for the nodes in a Dependency Views map.

If the node is a collapsed node or represents a cluster, the incidents, problems and change requests are for all the collapsed nodes.

1. Navigate to Dependency Views > View Map.
2. Click the down arrow icon (▼) next to a node or right-click a node on the map, to access the following menu items:

Table 521: Node Menu

<table>
<thead>
<tr>
<th>View Form</th>
<th>Displays the CMDB record of the selected CI in a new tab of the browser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Map</td>
<td>Reloads the view using the selected CI as the new root node, with the currently defined layout setting. This option does not display on the root node.</td>
</tr>
<tr>
<td>View Related Tasks</td>
<td>Displays all tasks or outages associated with the selected CI, including incidents, problems, change requests, and follow-on tasks. This option is always available, even if there are no tasks associated with the CI. This option does not appear on collapsed nodes.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>View Affected CIs</td>
<td>Shows a list of all tasks that have the CI listed as an Affected CI. This option is only visible when you access the view from the view icon in a task record's Configuration item field.</td>
</tr>
<tr>
<td>View Related Outages</td>
<td>Displays all outages involving the selected CI. This option only appears when there is an outage associated with the CI. This option does not appear on collapsed nodes.</td>
</tr>
<tr>
<td>Add Relationship</td>
<td>This option displays a dotted green line that you can drag to another CI to create a relationship link. A popup dialog allows you to define the relationship type.</td>
</tr>
<tr>
<td>Expand</td>
<td>Displays all CIs and components within a cluster node or a collapsed node. This option only appears if the node is a collapsed or cluster node. The number of additional icons to display is bound by the value of the glide.bsm.max_nodes property.</td>
</tr>
<tr>
<td>Collapse</td>
<td>Collapses all CIs and components within a cluster node or a collapsed node back to a single node. This option only appears if the node has been expanded using the Expand menu item.</td>
</tr>
<tr>
<td>Run Layout From Here</td>
<td>This option re-runs the chosen layout using the current node. Use this option to get a new or clearer view on the same map.</td>
</tr>
<tr>
<td>Load More</td>
<td>Starting at the selected icon, loads the next level of the map, past the setting of Max Levels. The number of additional icons to display is bound by the value of the glide.bsm.max_nodes property.</td>
</tr>
</tbody>
</table>

### Export a Dependency Views map

You can export a Dependency Views map to an image in PNG format.

1. Navigate to **Dependency Views > View Map**.
2. Configure the map view as you want the image to appear. The exported image displays the current view of the map.
3. Click the menu icon ().
4. Click **Export Image**.
5. Right-click the image and select **Save Image As**, **Print**, or any other menu option.
6. Click the "X" button to close the Export Image window.

View collapsed nodes in a Dependency Views map

Nodes can display in a collapsed format to avoid unnecessary clutter in large maps.

1. To expand a cluster or collapsed node, right-click the CI and select Expand from the context menu.
2. To collapse a cluster or a node with children, right-click the CI and select Collapse from the context menu.

Administer Dependency Views

Users with the admin role can control the appearance and behavior of Dependency Views by configuring map indicators, map related items, map icons, and menu actions.

Create or modify Dependency Views map indicators

Dependency Views maps use various icons next to a CI to indicate that the CI has related records such as alerts, outages, incidents and problems. These icons are called map indicators.

The default configuration includes map indicators for the following record types:

- Open incident.
- Unplanned current outage.
- Planned current outage, or an open problem.
- Current, planned, or recent change request.

You can filter out the display of affected CIs, alerts, current change requests, incidents and problems from the Filters menu. Also, you can create a map indicator to define additional record types, such as trouble sources for business service CIs. You can also modify an existing map indicator, for example to use a different color scheme or to alter the priority of a task. Affected CIs are based on the cmdb_ci field in the Task table and its extensions. Therefore, if you use custom tables to store CIs for incidents, problems and changes, the calculation of affected CIs is affected.

For more information on how map indicators are used to show tasks and outages in clusters and collapsed nodes, see Collapsed nodes in a Dependency Views map on page 2184.

1. Navigate to Dependency Views > Map Indicators.
2. Click New to create a new map indicator, or click the name of an indicator from the Table column to modify an existing map indicator.
3. Fill in the fields on the form, as appropriate.
Table 522: Map Indicator form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Name of the table represented by this map indicator.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The list shows only tables and database views that are in the same scope as the map indicator. Views are not supported, although included in the list.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the indicator.</td>
</tr>
<tr>
<td>Order</td>
<td>Priority order of the task. The highest priority task is the indicator with the lowest order number. When more than one indicator is present on a CI, the displayed color is the color associated with the highest priority task.</td>
</tr>
<tr>
<td></td>
<td>Additionally, a glyph on a CI displays the color indicator of the highest priority task attached to that CI.</td>
</tr>
<tr>
<td>Node color</td>
<td>This field is used for backward compatibility.</td>
</tr>
<tr>
<td>Icon</td>
<td>File name and path of the icon image file.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that allows you to enable or disable this record.</td>
</tr>
<tr>
<td>CMDB CI field</td>
<td>Name of the field on the selected table that contains the configuration item.</td>
</tr>
<tr>
<td>Start field</td>
<td>This field is used for backward compatibility.</td>
</tr>
<tr>
<td>End field</td>
<td>This field is used for backward compatibility.</td>
</tr>
<tr>
<td>Description</td>
<td>Name of the field on the selected table that contains the description of the configuration item.</td>
</tr>
<tr>
<td>Description</td>
<td>User-generated text to describe the indicator. Alphanumeric characters and spaces are valid for this field.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Condition builder that specifies when to apply this indicator. For example, a CI that has a current past outage is highlighted for 5 days. You can configure a condition to designate a different timeframe for what is considered to be current.</td>
</tr>
<tr>
<td>Active Dependencies</td>
<td>Check box to enable display of the indicator in a Dependency Views map.</td>
</tr>
<tr>
<td>Label</td>
<td>Text to display for the indicator.</td>
</tr>
<tr>
<td>Tooltip Label</td>
<td>The prefix portion of the tooltip (Tooltip Label : Tooltip info).</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tooltip Info</td>
<td>The suffix portion of the tooltip (Tooltip Label : Tooltip info).</td>
</tr>
</tbody>
</table>

4. Click **Submit** to enter a new map indicator. Click **Update** to modify an existing map indicator.

#### Create or modify Map Related Items

The Map Related Items module relates referenced CIs to one another, which allows them to be displayed in a Dependency Views map.

The base system configuration includes the following tables and relates them to items in the Computer [cmdb_ci_computer] and Server [cmdb_ci_server] tables.

- Disk [cmdb_ci_disk]
- Network Adapter [cmdb_ci_network_adapter]
- Database [cmdb_ci_database]

Some additional referenced CIs that can be related in this manner are file systems and running processes.

In the following example, computer nodes in the map are related to network adapter nodes if the Configuration Item field of the adapter records reference the specific CI node. Access or create a network adapter record from the Network Adapter related list in the cmdb_ci_computer record.

![Network Adapter](image)

The Dependency Views map for the *JEMPLOYEE-IBM computer shows the network adapter attached to the computer.
You can configure Dependency Views to display CIs that have no relationship record, but are related to other CIs by reference fields.

1. Navigate to **Dependency Views > Map Related Items**.
2. Click **New** to create a new related item, or click in the row of an existing CI to modify an existing map related item.
3. Fill in the fields on the form, as appropriate.
   
   See the Related Items form table.

4. Click **Submit** to enter a new map related item. Click **Update** to modify an existing map related item.

**Table 523: Related Items form**

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration item</td>
<td>CI that represents the base node or a CI in a table that extends the base node table. In the base system, the configuration item that represents the base node is Computer [cmdb_ci_computer], which includes all types of workstations and servers.</td>
</tr>
<tr>
<td>Related item</td>
<td>Table name of the related item. Only the cmdb_ci table and tables that extend it are displayed in the choice list.</td>
</tr>
<tr>
<td>Related field</td>
<td>Field that links this related item to the configuration item. In many cases, the appropriate value is automatically populated in the field after the first two fields are selected. Select the drop-down menu for additional options.</td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that allows you to enable or disable this record.</td>
</tr>
</tbody>
</table>

Create or modify Dependency Views map icons

You can upload new icons or modify existing icons to customize the icon displayed for a CI in Dependency Views maps.

Users with the admin or ecmdb_admin role can access the records in this table [ngbsm_icon] to upload new icons.

The icons used in Dependency Views maps are listed in the Map Icons module. Records in the Map Icons list are arranged by CI classes, such as cmdb_ci_linux_server. The path to the default image files is https://<instance name>.service-now.com/images/app.ngbsm/<image name.png>.

1. Navigate to Dependency Views > Map Icons.
2. Click New to create a new map icon, or click the name of an existing icon from the Name column to modify an existing icon.
3. Fill in the fields on the form, as appropriate.
   See the Map Icons form table.
4. Click Submit to enter a new icon. Click Update to modify an existing icon.

Table 524: Map Icons form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI Type</td>
<td>Label or the informal name of the CI table that this icon represents in the view.</td>
</tr>
<tr>
<td>Icon</td>
<td>Name of the icon.</td>
</tr>
<tr>
<td>URL</td>
<td>Path to the icon image using the following format: /image name.png Click the lock icon to enter a new path.</td>
</tr>
</tbody>
</table>

Create or modify Dependency Views menu actions

Best practices dictates that you modify a copy of the original menu action.

To modify an existing menu option, first you create a copy of the original menu action record, and then you modify the copy. This ensures that ServiceNow can update the record normally during the upgrade process and allows you to quickly restore the original menu option, if necessary.

1. To create a new menu option, navigate to Dependency Views > Map Menu Actions and click New.
   Fill in the fields on the form, as appropriate. See the Menu Action form table.
2. To modify an existing menu option, navigate to Dependency Views > Map Menu Actions.
3. Open the menu action you want to edit.
4. Right-click in the header and click Insert and Stay.
   This step creates a duplicate copy of the menu action and leaves it open for editing.
5. Change the name of the copied record to avoid confusion.
6. Modify the form fields as necessary and save the record.
7. Open the original record and disable it by clearing the Active check box.

### Table 525: Menu Action form

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Descriptive name that appears as the menu option.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that allows you to enable or disable this record.</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition that triggers the display of this menu option. If the condition evaluates to false the menu option does not display. Script is evaluated in JavaScript in the user's browser and does not have access to all the APIs that Business Rules do. For details on available parameters, see Condition Parameters.</td>
</tr>
</tbody>
</table>
| Item    | Map element for which the menu option displays. Valid values are:  
  • Canvas for the menu on the map background.  
  • Node for the menu on a CI.  
  • Relationship for the menu on a relationship link. |
| Order   | Physical location of the option in the menu. The option with the lowest order number appears first in the menu. All editable and custom options appear below the permanent menu options. |
| Script  | Script that is executed in the browser when the menu option is selected. Script is evaluated in JavaScript in the user's browser and does not have access to all the APIs that Business Rules do. |
| Type    | Menu action type being created, either a menu option or a menu separator. The menu separator is a single line. When the type is a separator, the Script field is ignored. |

---

**Legacy Business Service Management Map**

Legacy BSM is built on keyline technology.

To use the Legacy BSM application menu items, the bsm_legacy, admin, or the bsm_legacy_admin roles are required.
A ServiceNow business service is work or goods that are supported by an IT infrastructure. For example, delivering email service to an employee can require services such as email servers, web servers, and the work to configure the user's account. A business service management (BSM) map graphically displays the configuration items (CI) that support a business service and the relationships between the configuration items.

**Internet Browser Requirements and Limitations**

Microsoft Internet Explorer has these limitations and requirements when used with ServiceNow BSM maps:

- To access the BSM map using Internet Explorer 7 and 8, you must install Adobe Flash Player 10.1 or higher.

  **Note:** Geneva is the last release in which Internet Explorer 7 and 8 are supported for BSM maps.

- You cannot export images from the map with Internet Explorer 7.
- Map view images saved in a browser other than Internet Explorer 8 might not display properly in Internet Explorer 8.
- Internet Explorer 7 can only display the version numbers of saved views and not the images.

To use Firefox ESR - the latest version is required.

**Business service maps**

Many of the relationships in a BSM map are created through the discovery process.

You can also create, define, and delete CI relationships in the map. You can display the map from different perspectives and open specific records that relate to configuration items. The system refreshes the map automatically to reflect changes to the CMDB. Nodes on the map may represent one configuration item or a collection of configuration items grouped into a single icon. The following figure displays an overview of the actions that are available for BSM maps.
Roles

Users with the itil and ecmdb_admin roles can view BSM maps and perform all actions in the map, which includes accessing the map views and filters they have saved, both from the lists in the map and from the Map Filters module.

Installed with Legacy Business Service Management Map

What components are installed with BSM maps.

The following components are installed with BSM Maps:

Tables

Legacy BSM Maps includes the following tables.

**Table 526: Business Service Management Map tables**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSM Map Filter [bsm_map_filter]</td>
<td>Contains all saved BSM map filters.</td>
</tr>
<tr>
<td>BSM Map Icon [bsm_map_icon]</td>
<td>Contains the paths to all icon images used in a BSM map.</td>
</tr>
<tr>
<td>BSM Saved View [bsm_chart]</td>
<td>Contains the data for saved BSM snapshots.</td>
</tr>
<tr>
<td>BSM Related Item [bsm_related_item]</td>
<td>Stores which reference fields should be treated as relationships when building the map. This allows users to include CI's that are related via a reference field instead of a relationship.</td>
</tr>
</tbody>
</table>
Properties

To customize your map properties, navigate to BSM Map > Map Properties.
For properties with the type Color, use one of the following options:
- A valid HTML4 color name. The color names are case-insensitive.

The extended color set (X11 color keywords) typically work, as well.
- A RGB hex code, such as #003366
- A RGB decimal code, such as rgb(0,51,102)

Table 527: Business Service Management Map properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.bsm.map.style.text_color</td>
<td>The color of the text that appears under an unselected CI node. See above for color selection guidelines.</td>
</tr>
<tr>
<td>glide.bsm.map.style.selection_text_color</td>
<td>The color of the text that appears under a selected CI node. See above for color selection guidelines.</td>
</tr>
<tr>
<td>glide.bsm.map.style.font_size</td>
<td>The font size of the text that appears with a CI node. The default size is magnified for nodes with more connections and reduced for downstream nodes.</td>
</tr>
<tr>
<td>glide.bsm.map.style.selection_background_color</td>
<td>The background color of a selected CI node. This color is also used with a node’s Highlight Hierarchy option. See above for color selection guidelines.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.bsm.max_levels          | Maximum level depth from the root CI that can be initially displayed in Business Service Maps. Level depth is the graph distance between the root CI and a node. This value must be an integer.  
   • Type: Integer, valid values 1 to 10  
   • Default value: 5                                                                 |
| glide.bsm.map.style.font_family | The font family name used in the map text. If you designate a font that is not on your users' system, the browser substitutes another font and the text may not render as you expect.  
   • Type: Font name  
   • Default value: Arial                                                                 |
| glide.bsm.new_node_color      | The color for nodes that became viewable from the last expand operation. See above for color selection guidelines.  
   • Type: Color  
   • Default value: PaleGreen                                                                 |
| glide.bsm.too_many_children   | The maximum number of child nodes to display. Nodes are collapsed for the map to meet this limit.  
   • Type: Integer, valid values 1 or greater  
   • Default value: 10                                                                 |
| glide.bsm.color.affect_neighbors | The color of an affected neighbor node. When a node has a service issue, all the nodes that are dependent on that node are considered affected nodes. In the map, the affected nodes are parents or grandparents of the node with the service issue. See above for color selection guidelines.  
   • Type: Color  
   • Default value: Beige                                                                 |
| glide.bsm.max_nodes           | The maximum number of downstream nodes to retrieve from the database for a CI. If more nodes exist in the database, they are not displayed in the map.  
   • Type: Integer  
   • Default value: 1000                                                                 |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide bsm task_threshold</td>
<td>Change the CI's glyph color from orange to red when the number of tasks reaches this threshold.</td>
</tr>
<tr>
<td></td>
<td>• Type: Integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 3</td>
</tr>
<tr>
<td>glide bsm refresh interval</td>
<td>Seconds between each automatic reloading of troubles and tasks. The range is 1 to 3600.</td>
</tr>
<tr>
<td></td>
<td>• Type: Integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 3</td>
</tr>
<tr>
<td>glide bsm layout</td>
<td>The default layout for the BSM map. Options are Hierarchy, Radial, Grouping, and Force.</td>
</tr>
<tr>
<td></td>
<td>• Type: String</td>
</tr>
<tr>
<td></td>
<td>• Default value: Hierarchy</td>
</tr>
</tbody>
</table>

**Script includes**

BSM Maps includes the following script include.

**Table 528: Business Service Management Map script includes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeylinesBsmAJAX</td>
<td>Client callable AJAX for Keylines BSM maps.</td>
</tr>
</tbody>
</table>

**Business rules**

BSM Maps includes the following business rules.

**Table 529: Business Service Management Map business rules**

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Filter Name</td>
<td>BSM Map Filter [bsm_map_filter]</td>
<td>This rule ensures that a new filter name is not already in use. Filters must have unique names.</td>
</tr>
<tr>
<td>Insert/Update type</td>
<td>BSM Map Icon [bsm_map_icon]</td>
<td>This rule ensures that an icon type is unique for a category of CIs. For example, Linux servers should have only one icon type defined.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Validate BSM Map max child nodes</td>
<td>System Properties [sys_properties]</td>
<td>This rule ensures that the max child nodes property is a valid number greater than 0.</td>
</tr>
<tr>
<td>Validate BSM Map font size</td>
<td>System Properties [sys_properties]</td>
<td>This rule checks the font size for correct values and warns if the font size is too large and may cause rendering issues.</td>
</tr>
<tr>
<td>Validate Refresh Interval</td>
<td>System Properties [sys_properties]</td>
<td>This rule ensures that the refresh interval is a valid number between 1 and 3600, inclusive.</td>
</tr>
<tr>
<td>Validate BSM Map max levels</td>
<td>System Properties [sys_properties]</td>
<td>This rule governs the maximum level depth from the root CI that can be initially displayed in Business Service Maps. This rule ensures that the value is an integer between 1 and 10, inclusive.</td>
</tr>
<tr>
<td>Validate BSM Map max nodes</td>
<td>System Properties [sys_properties]</td>
<td>This rule checks max node value for valid numbers and minimum nodes shown.</td>
</tr>
<tr>
<td>Validate BSM Maps tasks threshold</td>
<td>System Properties [sys_properties]</td>
<td>This rule determines the threshold limit to change the task glyph from orange to red.</td>
</tr>
<tr>
<td>Increment view version</td>
<td>BSM Saved View [bsm_chart]</td>
<td>This rule determines the next version number for this saved BSM map.</td>
</tr>
<tr>
<td>Update Version</td>
<td>BSM Saved View [bsm_chart]</td>
<td>This rule verifies that a saved version of a BSM chart does not already exist.</td>
</tr>
</tbody>
</table>

**Administering Legacy Business Service Management maps**

ServiceNow Business Service Maps use color highlights to direct focus to a node, such as to indicate that a node has an outage. These highlights are called map indicators. Users with the admin role can control the appearance and behavior of business service management (BSM) maps by configuring map indicators, map related items, and map icons.

You can also edit or create options that appear on right-click menus in the map.

**Map Indicators**

The default configuration uses map indicators to differentiate between different types of tasks or outages. For example, a CI that has an open incident is highlighted in yellow.
Create or modify map indicators in a Legacy BSM map

You can create a map indicator to define additional record types, such as trouble sources for business service CIs. You can also modify an existing map indicator, for example to use a different color scheme or to alter the priority of a task.

To create or modify a map indicator:

1. Navigate to **Dependency Views > Legacy BSM > Map Indicators**.
2. Click **New** to create a new map indicator, or click the name of an indicator from the **Table** column to modify an existing map indicator.
3. Fill in the fields on the form, as appropriate.

---

**Table 530: Map Indicator Color Descriptions**

<table>
<thead>
<tr>
<th>Color</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow</td>
<td>CIs that have an open incident and all affected upstream CIs.</td>
</tr>
<tr>
<td>Red</td>
<td>CIs that have an open problem or a current, planned, or past outage.</td>
</tr>
<tr>
<td>Khaki</td>
<td>CIs that have open tasks.</td>
</tr>
<tr>
<td>Light Blue</td>
<td>CIs with a current, planned, or past change request.</td>
</tr>
</tbody>
</table>

---

**Table 531: Map Indicator Field form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Name of the table represented by this map indicator.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the indicator.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Order</td>
<td>Priority order of the task. The highest priority task is the indicator with the lowest order number. When more than one indicator is present on a CI, the displayed color is the color associated with the highest priority task. Additionally, a glyph on a CI displays the color indicator of the highest priority task attached to that CI.</td>
</tr>
<tr>
<td>Node color</td>
<td>Highlight color for a node when a task is attached from the table specified. This value can be expressed in hexadecimal code or color name.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that allows you to enable or disable this record.</td>
</tr>
<tr>
<td>CMDB CI field</td>
<td>Name of the field on the selected table that contains the configuration item.</td>
</tr>
<tr>
<td>Start field</td>
<td>Select the appropriate date/time field for the starting point of the <em>trouble</em> for this indicator. The tree picker displays only fields of the appropriate data type from the selected table. This data is not displayed on a map, but is used in the <em>issues pop-up</em> in the tree view of configuration items related to a Business Service.</td>
</tr>
<tr>
<td>End field</td>
<td>Select a date/time field for the end point of the <em>trouble</em> for this indicator. The tree picker displays only fields of the appropriate data type from the selected table. This data is not displayed on a map, but is used in the <em>issues pop-up</em> in the tree view of configuration items related to a Business Service.</td>
</tr>
<tr>
<td>Description field</td>
<td>Name of the field on the selected table that contains the description of the configuration item.</td>
</tr>
<tr>
<td>Description</td>
<td>User-generated text to describe the indicator. Alphanumeric characters and spaces are valid for this field.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Condition builder that specifies when to apply this indicator. For example, a CI that has a past outage is highlighted in red. You can configure a condition to designate that the red highlight no longer appears on the map one week after the outage is fixed.</td>
</tr>
</tbody>
</table>
4. Click **Submit** to enter a new map indicator or click **Update** to modify an existing map indicator.

Map related items module in Legacy BSM

The **Map Related Items** module relates referenced CIs to one another, which allows them to be displayed in BSM maps.

The base system configuration includes the following tables and relates them to items in the Computer [cmdb_ci_computer] and Server [cmdb_ci_server] tables.

- Disk [cmdb_ci_disk]
- Network Adapter [cmdb_ci_network_adapter]
- Database [cmdb_ci_database]

Some additional referenced CIs that can be related in this manner are file systems and running processes.

In the following example, computer nodes in a BSM map are related to network adapter nodes if the **Configuration Item** field of the adapter records reference the specific CI node. Access or create a network adapter record from the Network Adapter related list in the cmdb_ci_computer record.

![Network Adapter](image)

**Figure 586: BSM Related Items Example**

The BSM map for the *JEMPLOYEE-IBM* computer shows the network adapter attached to the computer.
Create or modify map related items in Legacy BSM

You can configure BSM maps to display CIs that have no relationship record, but are related to other CIs by reference fields.

To create or modify map related items:

1. Navigate to Dependency Views > Legacy BSM > Map Related Items.
2. Click New to create a new related item, or click the name of an existing CI from the Configuration Item column to modify an existing map related item.
3. Fill in the fields on the form, as appropriate.

Table 532: BSM Related Items form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Item</td>
<td>CI that represents the base node or a CI in a table that extends the base node table. In the base system, the configuration item that represents the base node is Computer [cmdb_ci_computer], which includes all types of workstations and servers.</td>
</tr>
<tr>
<td>Related Item</td>
<td>Table name of the related item. Only the cmdb_ci table and tables that extend it are displayed in the choice list.</td>
</tr>
<tr>
<td>Related Field</td>
<td>Field that links this related item to the configuration item. In many cases, the appropriate value is automatically populated in the field after the first two fields are selected. Select the drop-down menu for additional options.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that allows you to enable or disable this record.</td>
</tr>
</tbody>
</table>

4. Click **Submit** to enter a new map related item or click **Update** to modify an existing map related item.

Map icons in Legacy BSM

The icons used in the BSM maps are listed in the Map Icons module. Users with the admin or ecmdb_admin role can access the records in this table [bsm_map_icon] to upload new icons.

Records in the BSM Map Icons list are arranged by CI classes, such as cmdb_ci_linux_server. The path to the default image files is https://<instance name>.service-now.com/images/keylines_bsm_map/<image name.png>.
Create or modify map icons in Legacy BSM
You can upload new icons or modify existing icons to customize the icon displayed for a CI.

To create or modify a map icon:

1. Navigate to **Dependency Views > Legacy BSM > Map Icons**.
2. Click **New** to create a new map icon, or click the name of an existing icon from the **Name** column to modify an existing icon.

3. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the icon.</td>
</tr>
<tr>
<td>Type</td>
<td>Label or friendly name of the CI table this icon represents on the map.</td>
</tr>
<tr>
<td>Icon Path</td>
<td>Path to the icon image using the following format: <code>images/keylines_bsm_map/&lt;image name&gt;.png</code> Click the lock icon to enter a new path.</td>
</tr>
</tbody>
</table>

4. Click **Submit** to enter a new icon or click **Update** to modify an existing icon.
   The system creates or updates the record.

Menu actions in Legacy BSM
Menu actions are options that appear in the context menu that displays when you right-click a map node or relationship link in a BSM map. The actions that appear in the menu differ depending on pre-defined conditions. The Menu Actions module allows you to customize some of the menu options that are provided with the application, or you can create new menu options.

**Note:** Creating and modifying menu actions is an advanced procedure that requires scripting knowledge.
Protected Menu Options

Some right-click menu options are essential to the application and cannot be customized. Permanent options appear above the editable or custom options in the menu.

![Protected Menu Options](image)

Figure 588: Protected Menu Options

Node Menus

The following menu options on CI nodes are not editable:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Opens the record for the selected CI.</td>
</tr>
<tr>
<td>Redraw layout</td>
<td>Redraws the map with the selected CI as the focus.</td>
</tr>
<tr>
<td>Highlight hierarchy</td>
<td>Highlights the CIs that depend on the selected CI, including those CIs on which the selected CI is dependent.</td>
</tr>
<tr>
<td>Expand this node</td>
<td>Displays all CIs and components within a cluster node or a collapsed node. This option only appears if the node is a collapsed or cluster node.</td>
</tr>
<tr>
<td>Set as CI</td>
<td>Replaces the value in the original task's reference field with the name of the selected CI. This action only appears for maps accessed from task records.</td>
</tr>
</tbody>
</table>

Relationship Menus

The following menu options on a relationship line are not editable:
### Table 535: Relationship Menu Option Descriptions

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit connection</td>
<td>Opens the Edit Link Configuration dialog box. Select a new relationship for this link from the choice list.</td>
</tr>
<tr>
<td>Remove connection</td>
<td>Deletes a relationship. The system deletes the relationship without prompting for confirmation.</td>
</tr>
<tr>
<td>Redraw layout</td>
<td>Returns the focus to the most recently selected CI.</td>
</tr>
</tbody>
</table>

*Create or modify menu actions in Legacy BSM*

To change an existing menu option, copy the original menu action record, and then modify the copy. This ensures that ServiceNow can update the record normally during the upgrade process and allows you to restore the original menu option quickly, if necessary.

To create a new menu option:

1. Navigate to **Dependency Views > Legacy BSM > Menu Actions** and click **New**.
2. Fill in the fields on the form, as appropriate.

Table 536: BSM Menu Action form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Descriptive name that appears as the menu option.</td>
</tr>
<tr>
<td>Item</td>
<td>Map element for which the menu option displays. Valid values are <strong>Node</strong> for the menu on a CI and <strong>Relationship</strong> for the menu on a relationship link.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Menu action type being created, either a menu option or a menu separator. The menu separator is a single line. When the type is a separator, the <strong>Script</strong> field is ignored.</td>
</tr>
<tr>
<td>Order</td>
<td>Physical location of the option in the menu. The option with the lowest order number appears first in the menu. All editable and custom options appear below the permanent menu options.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box that allows you to enable or disable this record.</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition that triggers the display of this menu option. If the condition evaluates to false the menu option does not display.</td>
</tr>
<tr>
<td>Script</td>
<td>Script that is executed in the browser when the menu option is selected.</td>
</tr>
</tbody>
</table>

3. **To change an existing menu option:**
   a) Navigate to **BSM Maps > Menu Actions** and open the menu action you want to edit.
   b) Right-click in the header and click **Insert and Stay**. This creates a duplicate copy of the menu action and leaves it open for editing.
   c) Change the name of the copied record to avoid confusion.
   d) Modify the form fields as necessary and save the record.
   e) Open the original record and disable it by clearing the **Active** check box.

**Condition parameters in Legacy BSM**

The **Condition** field contains a boolean expression that evaluates to true or false. If the condition is true or if there is no condition, the specified option appears in the menu when you right-click a CI or a relationship link. When you select the option from the menu, ServiceNow executes the associated script.

---

**Note:** The usual regular expression conventions are valid in the condition field, such as `!` for NOT, `&&` for AND, and `||` for OR.

---

![Figure 589: Condition field](image)
### Table 537: Common Elements for Building a Condition

<table>
<thead>
<tr>
<th>Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item.d</td>
<td>Node or reference link's data on which you performed the right-click action.</td>
</tr>
<tr>
<td>item.d.label</td>
<td>Label of the node.</td>
</tr>
<tr>
<td>item.d.ci_type</td>
<td>CI's type (table), such as <code>cmdb_ci_service</code>.</td>
</tr>
<tr>
<td>item.d.name</td>
<td>Name of CIs or relationships.</td>
</tr>
<tr>
<td></td>
<td>• <strong>CIs</strong> CI's type name or the table label, such as <code>Business Service</code>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Relationships</strong> Relationship's type name, such as <code>Depends on::Used by</code>.</td>
</tr>
<tr>
<td>item.c.status.tablename</td>
<td>Status of the node in the table <code>tablename</code>. Replace <code>tablename</code> with the name of a map indicator table, as listed in the Map Indicators module.</td>
</tr>
<tr>
<td>item.d.parent_relationship</td>
<td>Parent relationship on a relationship link.</td>
</tr>
<tr>
<td>item.d.child_relationship</td>
<td>Child relationship on a relationship link.</td>
</tr>
<tr>
<td>item.d.location</td>
<td>Location of the CI, such as New York.</td>
</tr>
<tr>
<td>item.d.manufacturer_name</td>
<td>Name of the CI's manufacturer, such as Dell Inc.</td>
</tr>
</tbody>
</table>

The following table provides examples of valid conditions.

### Table 538: Valid Conditions for Condition Parameters

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item.d.is_collapsed</td>
<td>The node is a collapsed node.</td>
</tr>
<tr>
<td>item.d.is_cluster</td>
<td>The node is a cluster node.</td>
</tr>
<tr>
<td>item.d.status.incident</td>
<td>The CMDB has an incident for that CI in the incident table.</td>
</tr>
<tr>
<td>item.d.status.cmdb_ci_outage</td>
<td>The CMDB has an outage for that CI in the <code>cmdb_ci_outage</code> table.</td>
</tr>
<tr>
<td>item.d.parent_relationship == &quot;Depends on&quot;</td>
<td>This relationship link is a depends on relationship.</td>
</tr>
</tbody>
</table>

*Script parameters in Legacy BSM*

Menu action scripts are executed on the client when a user clicks the menu option. You can use the same building blocks in scripts as in conditions. Menu action scripts do not function on separators.

These are some additional, useful expressions for scripts:
Table 539: Script Parameters

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item.id</td>
<td>The sys_id of the CI node or relationship link.</td>
</tr>
<tr>
<td>item.d.source</td>
<td>The sys_id of the relationship's parent or child.</td>
</tr>
<tr>
<td>item.d.target</td>
<td>The sys_id of the relationship's parent or child.</td>
</tr>
<tr>
<td>item.d.label</td>
<td>The name of the CI node, such as IronMail-SD-02.</td>
</tr>
<tr>
<td>item.d.location</td>
<td>The sys_id of the CI node's location.</td>
</tr>
<tr>
<td>item.d.location_name</td>
<td>The full address of the location, such as 4616 Clairemont Drive, North Clairemont, San Diego CA.</td>
</tr>
<tr>
<td>item.d.manufacturer_id</td>
<td>The sys_id of the CI's manufacturer.</td>
</tr>
</tbody>
</table>

Use a Legacy BSM map

How to use a Legacy Business Service Management (BSM) map.

A business service management (BSM) map has one starting point, called the root CI or root node of the map. The root CI is highlighted with a circle around it. The maps can show both upstream and downstream dependencies for the root CI. By default the BSM map displays 3 levels, both upstream and downstream relationships, and collapses all clusters. Administrators can configure the number of levels displayed.

Use the layout controls to display map elements in different configurations for easier management. Use the filter panel to display fewer levels or to filter out elements you don’t want to see, then save the filter for use later. Draw new relationships between elements or edit existing relationships.

In a BSM map, icons and glyphs indicate if a CI has an active, pending issue. You can investigate the tasks that are connected to a CI to get more details. The map collapses and expands clusters to make them easier to view.

Access a Legacy BSM map

When you access a Legacy BSM map from one of the view options, the map is centered on the root CI, and displays the layout and number of levels defined in Map Properties.

Administrators can configure these settings.

When you access the map from a saved view, the map opens using the saved properties.

To access a BSM map, navigate to Dependency Views Legacy BSM and open one of these modules:

Table 540: BSM map modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Map in New Tab</td>
<td>Opens the map in a new, full screen tab without the application navigator.</td>
</tr>
<tr>
<td>View Map</td>
<td>Opens the map in the content pane of the current tab.</td>
</tr>
</tbody>
</table>
### Saved Views

Opens a view of a map that you previously saved. Click a number in the Version column, and then click the BSM map icon ( ![BSM map icon](link) ).

**Note:** Adobe Flash Player is required to access BSM maps using Microsoft Internet Explorer versions 7 and 8.

---

**Manage a Legacy BSM map**

You can manage the map using controls on the map and controls on the CI.

#### Table 541: Map controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI Search Field</td>
<td>Select a CI from the search field above the map. Alternatively, you can start typing to have the auto-complete feature present a list of CIs that match your partial value. The search field displays the name of the root CI shown in the map and not the currently focused node. When you return to the map from another node, the system restores the last map viewed, using the default filter and layout settings.</td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Magnifier</td>
<td>Use the zoom control to magnify the map. Use the direction arrows to move the entire map, or the selection tool to move one or more elements. Click the selection tool again to change it into the grab tool for moving the entire map. Click the circle in the center of the direction arrows to center the image.</td>
</tr>
</tbody>
</table>
Control | Description
--- | ---
Layout | Select a layout for the map elements from the choice list. You can set a default layout through a *map property*.  

<table>
<thead>
<tr>
<th>Choice</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy</td>
<td>Displays elements in a tree pattern according to their upstream and downstream relationships. This is the default value for the initial display of the map.</td>
</tr>
<tr>
<td>Radial</td>
<td>Displays elements in a radial pattern according to their upstream and downstream relationships.</td>
</tr>
<tr>
<td>Grouping</td>
<td>Groups elements according to their CI type.</td>
</tr>
<tr>
<td>Force</td>
<td>Centers elements around the parent CI, regardless of upstream or downstream relationships.</td>
</tr>
</tbody>
</table>

Collapse | Collapse an expanded node. The Collapse button is visible when an expand command adds new nodes to the graph.  
If you expand multiple times in a row, the Collapse button only goes back one collapse action. Collapse memory is not saved to the database, and therefore is available in the current session only. |

Export Image | *Save a copy* of the current map in the PNG format. |

Print Map | Print the current map or save the map as a PDF file. |
<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save View</td>
<td>Save the current map view in the database. The saved view preserves node placement and filter conditions at the point in time of the save command. Each view is assigned an automatically incrementing version number in the thumbnail view. In addition, all saved views are available from the Saved Views module. A message notifies the user that the current view has been saved. Views and versions are saved based on the current root CI.</td>
</tr>
<tr>
<td>Load View</td>
<td>Load a previously saved view of the BSM map. The selections available are thumbnail views of your saved map configuration for the root node. Selecting a thumbnail will load both the node placement and the filter conditions that were previously saved. The root node is highlighted with a circle around it. Saved views are loaded from the database.</td>
</tr>
<tr>
<td>Full Screen</td>
<td>Toggle the display of the map in and out of full screen mode. The full screen view is not supported on Internet Explorer. This option is available in the View Map in New Tab view.</td>
</tr>
<tr>
<td>Filter Panel</td>
<td>Toggle between opening and closing the filter panel.</td>
</tr>
</tbody>
</table>

Right-click a CI node to access these controls.
Table 543: CI node controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>View CI name</td>
<td>Displays the record of the selected CI in a pop-up window. You can also double-click the CI to open this window.</td>
</tr>
<tr>
<td>Redraw layout</td>
<td>Redraws the map using the selected CI as the new root node, using the currently defined layout setting.</td>
</tr>
<tr>
<td>Expand this node</td>
<td>Expands the collapsed nodes in the map that are associated with the selected CI. The CI type, manufacturer, location, and relationship filters are updated to reflect the nodes on the expanded map.</td>
</tr>
</tbody>
</table>
| Highlight hierarchy   | Highlights related nodes upstream and downstream from the node with the color designated in the map property. The highlight appears as the node background color. The highlight also appears on nodes that are dimmed due to selected filters. The node is related if it meets these criteria:  
  • It is an ancestor, or a direct ancestor of an ancestor, of the node.  
  • It is a child, or a direct child of a child, of the node.  
  The children of ancestors and the ancestors of children are not included. |
| Show related tasks    | Displays all tasks or outages associated with the selected CI, including incidents, problems, change requests, and follow-on tasks. This option is always available, even if there are no tasks associated with the CI. |
| Show related outages  | Displays all outages involving the selected CI. This option only appears when there is an outage associated with the CI.               |
| Add affected CI       | Adds the selected CI to the Affected CI related list in an incident, problem, or change request. This option is only visible when you access the map from the map icon in a task record's Configuration item field. You might need to configure the incident, problem, or change form to display the Affected CI related list. |

Filter a Legacy BSM map

Use the filter to control which elements of the map are highlighted and to save versions of a filter for later use.

Open or close the filter panel with the Filter Panel button in the map's header bar.
Filter groups and controls in the filter panel are contained within collapsible strips. Click a strip to expand or collapse it. Within each strip, you can select individual items or use the All check box to select all items in that strip.

Figure 590: BSM filter panel
Table 544: BSM filter panel

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save Filter</td>
<td>Saves the current filter settings. The system asks you to provide a unique name for the filter and saves it to the BSM Map Filter [bsm_map_filter] table. ServiceNow overwrites an existing filter if it has the same name as the one you are saving. To view saved filters, navigate to <strong>BSM Map &gt; Map Filters</strong>.</td>
</tr>
<tr>
<td>Load Filter</td>
<td>Allows you to select from a list of saved filters. When you select a saved filter, the system applies the settings to the map as appropriate for the filter groups and subgroups that are present.</td>
</tr>
<tr>
<td>Reset Filter</td>
<td>Resets the filter to its default state. This action selects all check boxes in the group strips and resets the Stream direction and Level depth options to their default values.</td>
</tr>
</tbody>
</table>

Direction and level in Legacy BSM maps

By default the Legacy BSM map displays three levels of upstream and downstream relationships.

To change the number of levels that are displayed by default, navigate to **Dependency Views > Legacy BSM > Map Properties** and edit the Maximum level depth from the root CI that can be initially displayed in Business Service Maps property.

When you load a new CI or expand a node, the system resets the map to the configured level depth.

Use these map controls to change the view of the levels as appropriate:

Table 545: Map controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream direction</td>
<td>Direction of relationships to display. Select from Upstream, Downstream, or Both directions. The default map displays nodes in both directions.</td>
</tr>
</tbody>
</table>
### Control

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
</table>
| Level depth | Number of relationship levels to highlight for the current node. Select **Show all** to set the depth to the number of levels that are in your CI configuration. When you perform one of the following tasks, the system isolates that node and displays the selected number of levels in the specified direction from that node:  
  - Click a node  
  - Load a new node  
  - Change CIs using the search box  
  - Expand a node |

### CI filters by group in Legacy BSM maps

Expand a filter group strip to display the subgroups. For example, the CI by Type group contains subgroups such as Database, UNIX Servers, and Network Gear.

Select a check box to filter the map to show CIs from that subgroup and their relationships in the map. Clear a check box to dim that subgroup of CIs in the map. Scroll long lists of subgroups.

By default all options in each group are selected. If no manufacturer or location is defined in the database, the Other check box is the only option and is selected. Clear a check box to dim those CIs or relationships. Select the All check box to highlight all CI subgroups or clear the check box to dim all subgroups. For example, you might want to clear all check boxes in a long list and then reselect only one or two subgroups to highlight. The All control only works for the current group. Make sure a group filter shows the desired view before you switch groups, since this can affect what you see in subsequent views. For example, if you dim database servers in the CI by Type group and then change to the Location group, you cannot highlight database servers in any location.

### Tasks by type and date in Legacy BSM maps

Click the Tasks by Type & Date strip to show the tasks by type for the CIs on the map.

Click the **Tasks by Type & Date** strip to show the tasks by type for the CIs on the map. Clear the check box for a task type to subtract those tasks from the total displayed in the glyph. When the last task on the CI is hidden, the glyph disappears.

Select from these task types:

- Open Incidents
- Open Problems
- Open Changes
- Open Requests
- Audit Tasks

Select a time period for the task filter from the Date opened field:

- All times
- In last 7 days
- In last 30 days
- In last 90 days
By default, an orange task glyph on a CI indicates the number of open tasks that are associated with that CI. A red glyph indicates that the CI has reached its task threshold. These properties are configurable.

View map information glyphs in a Legacy BSM map

When a CI has a service incident or a task associated with it, a glyph appears on the map. You can change the displayed data point for a class of icons in the Map Properties page.

Incidents, or troubles, and tasks share one glyph in the lower left corner of the CI. Each glyph has a color and a symbol. The symbol may be a number that represents a metric or count from other information. The following rules apply:

- If there are tasks but no troubles, the glyph color is orange. The glyph color turns red if the number of tasks reaches the configured threshold (default is 3).
- If there are troubles, with or without tasks, the glyph uses the color set in the Map Indicator module.
- If there are troubles with no tasks, the symbol is an exclamation mark (!).

You can change the color of the glyph in the Map Indicators module.

**Note:** For information on how glyphs and color highlighting are used with clusters and collapsed nodes, see Collapsed Elements in a BSM Map.

The glyphs that appear on the CI icons provide the following information.

<table>
<thead>
<tr>
<th>Glyph</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Cluster node" /></td>
<td>Cluster node</td>
<td>Shows the number of elements in a collapsed cluster node. To expand the node, right-click the CI and select <strong>Expand this</strong> from the context menu.</td>
</tr>
<tr>
<td><img src="image" alt="Collapsed node" /></td>
<td>Collapsed node</td>
<td>Shows the number of CIs in a collapsed node. To expand the node, right-click the CI and select <strong>Expand this</strong> from the context menu.</td>
</tr>
<tr>
<td><img src="image" alt="Task threshold exceeded" /></td>
<td>Task threshold exceeded</td>
<td>Appears when the configured threshold for number of tasks is reached. Indicates the total number of tasks associated with the CI. To view the tasks associated with this CI in a pop-up window, right-click the CI and select <strong>Show Related Tasks</strong> from the context menu.</td>
</tr>
<tr>
<td>Glyph</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><img src="image" alt="2" /></td>
<td>Task count</td>
<td>Shows the number of open tasks associated with this node. To view the tasks associated with this CI in a pop-up window, right-click the CI and select <strong>Show Related Tasks</strong> from the context menu. Alternatively, point to the glyph to display a pop-up window with detailed information about tasks associated with this CI.</td>
</tr>
<tr>
<td>![!]</td>
<td>Trouble with no tasks</td>
<td>Indicates that there are troubles but no tasks. To view the incidents associated with this CI in a pop-up window, right-click the CI and select <strong>Show Related Outages</strong> from the context menu.</td>
</tr>
<tr>
<td>![+]</td>
<td>Add relationship</td>
<td>Creates relationships between CIs.</td>
</tr>
</tbody>
</table>

**Export a Legacy BSM map**

You can export a Legacy BSM map to an image in the PNG format.

1. In the Legacy BSM map page, click **Export Image**. A thumbnail image of the map appears.
2. Right-click the thumbnail and select **Save Image As**.
3. Save the PNG image to the local drive or to a location on the network.

Manage relationships in a Legacy BSM map

You can create new relationships between CIs in the BSM map. You can also edit or delete existing relationships. All changes to relationships are automatically updated in the CMDB.

View a relationship in a Legacy BSM map

To identify a current relationship, point to the relationship link to activate the label.

To make connectors more visible in congested areas of the map:

1. Select the connector.
2. Click and drag the blue spot that appears to make the line curve.
Create a relationship in a Legacy BSM map

How to create a relationship.

The following rules apply to creating a relationship between nodes on the map:

• You cannot add a relationship to or from a CI if the node is collapsed.
• You cannot create a relationship from a CI to itself.
• You cannot create a relationship between an active node and a dimmed node.
• You can add a relationship to or from a cluster node.

**Note:** The BSM map only displays the upstream and downstream relationships from the root node selected as the focus of the map. It does not show any direct relationships between a child of the root node and a parent of the root node. If you add such a relationship and redraw the map with the same root node, the relationship line does not appear on the map. However, if you focus the map on a different root node directly upstream or downstream of the new relationship, the connector appears where you created it.

1. Select the CI you want to use as the parent in the relationship.
   The green plus glyph appears on the icon for the CI.

2. Click the plus glyph and drag a new connector to the child CI.
   A link configuration dialog box appears.
3. Select a relationship from the choice list and click **OK**.
   The map displays the new connector and label.

![Image of relationship between web application and application]

**View a change to a relationship in a Legacy BSM map**

Changes that involve relationships between CIs are automatically recorded in the configuration record for each CI.

You can view the changes in the Audit Records section as relationship removed or relationship added. You may need to configure the form to display the audit records.

![Image of audit records showing relationship changes]

**Figure 591: BSM relationship changes**

1. Right-click the CI node.
2. Select the **View** option on the context menu to open the configuration record for the CI.

**Edit a relationship in a Legacy BSM map**

You cannot right-click to edit the relationship link if the relationship type is not defined in the CI Relationship Type \[cmdb_rel_type\] table.

To edit the type of relationship between two CI nodes if the relationship type is defined in the cmdb_rel_type table:

1. Right-click the connector.
2. Select **Edit connection** from the context menu.
   The Edit Link Configuration dialog box appears. The current relationship is selected.
3. Select a new relationship from the choice list.
4. Click **OK**.

**Delete a relationship in a Legacy BSM map**

How to delete a relationship.

Note: When you delete a relationship between two nodes, the application does not require confirmation. The relationship is immediately removed from the CMDB.

To delete a relationship, right-click the connector and select **Remove connection** from the context menu.

You cannot delete the relationship if the relationship type is not defined in the CI Relationship Type [cmdb_rel_type] table.

**Collapsed elements in a Legacy BSM map**

The ServiceNow BSM map can display clusters and child nodes alongside individual CI nodes.

Aspects of the map can differ in these cases, for example the nodes can display in a collapsed format to avoid unnecessary clutter in large maps.

A cluster is an organized set of computer CIs that work together as a single system. Each node in the cluster represents a CI, typically a server, that can have referenced hardware, such as disks and network adapters. Clusters are CIs in the Cluster [cmdb_ci_cluster] table and can have tasks assigned to them. ServiceNow automatically collapses all clusters it encounters when it creates a BSM map.

A collapsed node collects all the children of a CI that are in the same class into a single node when the number of children in that class exceeds a configurable threshold. Collapsed nodes contain CIs but are not CIs themselves and cannot have tasks assigned to them.

**Annotation in a Legacy BSM map**

Icons for clusters and collapsed nodes are highlighted in blue and display a counter glyph in the upper left that indicates the number of nodes contained in the collection.

The system searches through all the component nodes in a cluster CI or collapsed node looking for tasks, outages, and trouble, such as incidents, problems, or change requests. This search evaluates only the number of levels that are displayed in the diagram. A glyph appears on the lower left edge of the icon if any tasks, incidents, or outages are associated with nodes in the collection. This glyph displays the number of tasks assigned to nodes within the collection and appears in the color of the highest priority task.

Note: All incidents, problems, changes, and tasks are displayed in a single list. The system uses map indicators to establish a priority (order number) that determines which task indicator colors are displayed in task glyphs. Outages are displayed in a different list.
By default, the system automatically updates the map every 30 seconds. To change the refresh interval, navigate to **BSM Maps > Map Properties** and edit the **Seconds between each automatic reloading of troubles and tasks** property. The system also updates changes to the CIs in the map when the map is reloaded.

**View a collapsed object in a Legacy BSM map**

- To expand a cluster or collapsed node, right-click the icon and select **Expand this** from the context menu.
- To collapse a cluster or a node with children, click **Collapse** in the map header bar.

The following behaviors apply to collapsible nodes:

- The collapse option only collapses the most recently expanded node.
- The map view retains the last filter settings when the collection is collapsed.
- If the cluster itself contains tasks, the count that displays in the task glyph includes the total count for tasks on the cluster CI. The color of the glyph is the color of the highest priority task in the cluster.
- When a cluster is expanded, the counter glyph disappears.
- If the cluster CI has no tasks associated with it, the task glyph disappears.
- When a collapsed node is expanded, the icon disappears until you collapse the nodes again.

![Figure 592: Collapsed Cluster Node with Tasks](image)
You can configure the color and order number for tasks, outages, and troubles, such as incidents, in the Map Indicators module. The order number establishes priority when the system displays indicators for multiple tasks.

**Tasks**

A collapsed collection with tasks assigned to its nodes displays an orange glyph. If the total number of tasks within the collection meets or exceeds the configured maximum in the threshold property, the glyph color is red. These colors are not configurable.

**Outages**

A collapsed collection indicates when its nodes have current, planned, or past outages. In the default configuration, the color for any outage is red, and current and planned outages have the highest priority. Next in priority are incidents and problems, which means their configured color displays over past outages. Outages are not included in the activity count shown in the task glyph.

**Troubles**

If the collapsed collection contains one node with trouble, the system displays a glyph in the color defined for that map indicator. If there are multiple trouble records, the glyph shows the color of the indicator with the lowest order number. For example, in a default configuration, if there are incidents and problems within the collection, the glyph uses the map indicator color for incidents, which has a lower order number than problems. If the nodes in the collection have a problem, a change request, and some tasks, the glyph shows the color of the problem, which has a lower order number than the others.
Managing when past outages appear

To manage when past outages appear for a CI:

1. Navigate to **Dependency Views > Legacy BSM > Map Indicators**.
2. Click the reference icon for the **outage_past** indicator.
3. In the outage record, configure one of the following:
   - Clear the **Active** check box. This prevents past outages from displaying on the BSM map at any time.
   - Edit the condition statement to limit the length of time past outages display on the map.
4. Click **Update**.
Component nodes

An expanded collection shows all nodes that satisfy the level depth configured for the map.

Node icons show glyphs for tasks, trouble, or outages associated with them, and neighbor nodes affected by trouble are highlighted in the appropriate affected node color. The cluster node itself is not displayed after you expand it.

Figure 595: Expanded Cluster With Trouble in Nodes
Figure 596: Expanded Nodes With an Outage

**Visual Task Boards**

Visual Task Boards (VTB) transform the navigation of lists and forms into an interactive graphical experience.

Visual Task Boards allow you to view and update multiple task records, which appear as *cards* that can be moved between *lanes*. An *activity stream* on the board displays recent activity so you can easily track changes to tasks. You can add task cards from any table that extends Task to intuitively and easily track updates and edit records directly from the board. Any user can use task boards, regardless of role, though access control rules (ACLs) may limit which cards each user can see. The Visual Task Board interface provides a graphic-rich environment suited for managing and collaborating on records. For example, a support manager might create a board for her team to track their assigned incidents by state in real time.

This video demonstrates how Visual Task Boards work.

**Note:** UI16 or UI15 is required to use Visual Task Boards.

The following podcast offers additional information on the use of Visual Task Boards.
Supported browsers for Visual Task Boards

The system supports Visual Task Boards on the latest version or service pack of the following browsers.

- Firefox version 31 and up
- The latest public release of Chrome
- Safari version 6.1 and up
- Internet Explorer version 10 and up
  - Edge mode is supported.
  - Compatibility mode is not supported.
  - Internet Explorer 11 is susceptible to memory leaks, which may impact performance, especially in Windows 7.

Scripting support

Visual task boards do not interact with client scripts. Use a UI policy or a data policy instead.

Board structure

There are different types of task boards for different kinds of task management. All types of boards share the same overall structure.

<table>
<thead>
<tr>
<th>Board type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freeform boards</td>
<td>Display any kind of task record, including personal tasks. Members of freeform boards can add and remove task cards and lanes.</td>
</tr>
<tr>
<td>Flexible boards</td>
<td>Display tasks that match the configured filter against a particular table. Members of flexible boards can add task cards, which are removed automatically when the tasks no longer match the filter conditions. Members can define custom lanes, similar to a freeform board.</td>
</tr>
<tr>
<td>Guided boards</td>
<td>Display tasks that match the configured filter against a particular table, like flexible boards. Members of guided boards can add task cards, which are removed automatically when the tasks no longer match the filter conditions. Guided board lanes correspond to field values and cannot be edited in most cases.</td>
</tr>
</tbody>
</table>

The icon beside the board name on the My Task Boards page identifies the type of board. Freeform boards appear with a grid of four squares ( ); flexible boards appear with a vertical line beside two squares ( ); guided boards appear with two vertical lines ( ).
Figure 597: My Task Boards screen

All boards have these elements:
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick panel</td>
<td>Displays labels and users associated with the board. Board members can use the quick panel to filter cards or to quickly label or assign tasks. Members can also configure what appears in the quick panel.</td>
</tr>
<tr>
<td>Lanes</td>
<td>Organize cards on a board into vertical groups. These groups often represent the status of the task, such as To Do, Doing, and Done. Each board is composed of one or more lanes. When using a guided board, each lane represents a possible field value. For example, a board on the Incident table can display one lane for each State value such as New, Active, or Resolved. Users can move cards from one lane to another to update the task that card represents.</td>
</tr>
<tr>
<td>Cards</td>
<td>Represent individual tasks. Users can add comments, attachments, and labels to cards and assign users to cards. Each card is tied to a task record; updating one immediately updates the other. For freeform boards, each card represents a personal task. For flexible and guided boards, each card represents a record from the list that board was created from.</td>
</tr>
<tr>
<td>Activity stream</td>
<td>Displays a history of changes to cards on the current board, as well as tabs for member and label administration.</td>
</tr>
</tbody>
</table>
Activate Visual Task Boards

Visual Task Boards are active by default on new instances. For instances upgrading from a previous version, the Visual Task Boards plugin (com.glide.ui.vtb) must be activated.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Installed with Visual Task Boards

Several types of components are installed with Visual Task Boards.

Tables installed with Visual Task Boards

Visual Task Boards add the following tables.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board Member [vtb_board_member]</td>
<td>Stores a record for each member of each task board.</td>
</tr>
<tr>
<td>Private Task [vtb_task]</td>
<td>Stores a record for each personal task on each freeform board.</td>
</tr>
<tr>
<td>Visual Task Board [vtb_board]</td>
<td>Stores a record for each task board.</td>
</tr>
<tr>
<td>Visual Task Board Card [vtb_card]</td>
<td>Stores a record for each task card on each task board.</td>
</tr>
<tr>
<td>Visual Task Board Card History [vtb_card_history]</td>
<td>Stores records for certain task card changes, such as lane or assignee changes. Each record indicates the previous value and the new value.</td>
</tr>
<tr>
<td>Visual Task Board Lane [vtb_lane]</td>
<td>Stores a record for each lane on each task board.</td>
</tr>
</tbody>
</table>

Properties installed with Visual Task Boards

Visual Task Boards add the following properties.
Note: To open the System Property [sys_properties] table, enter `sys_properties.list` in the navigation filter.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vtb.board.upload_limit</td>
<td>Sets the maximum file size in megabytes allowed for each task card attachment.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: 10</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: System Property [sys_properties] table</td>
</tr>
</tbody>
</table>

Script includes installed with Visual Task Boards

Visual Task Boards add the following script includes.

<table>
<thead>
<tr>
<th>Script include</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisualTaskBoards</td>
<td>Provides general utilities for Visual Task Boards.</td>
</tr>
<tr>
<td>VTBBoardSecurity</td>
<td>Provides security utilities to control board access.</td>
</tr>
<tr>
<td>VTBTaskSecurity</td>
<td>Provides security utilities to control task access.</td>
</tr>
</tbody>
</table>

Client scripts installed with Visual Task Boards

Visual Task Boards add the following client scripts.

<table>
<thead>
<tr>
<th>Client script</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Kanban Choice</td>
<td>Visual Task Board [vtb_board]</td>
<td>Adds the <strong>Kanban Board</strong> option to the Lane field. This option indicates that the task board is a flexible board.</td>
</tr>
</tbody>
</table>

Business rules installed with Visual Task Boards

Visual Task Boards add the following business rules.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add members to live group</td>
<td>Board Member [vtb_board_member]</td>
<td>If a Connect Chat conversation is created for a task board, this business rule adds the board members as conversation members.</td>
</tr>
</tbody>
</table>
### Business rule

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sync Members With VTB</td>
<td>Live Group Member</td>
<td>If a Connect Chat conversation exists for a task board and a member is added to the conversation, this business rule adds the conversation member as a board member.</td>
</tr>
<tr>
<td>Update Board on Card Change</td>
<td>Visual Task Board Card</td>
<td>Triggers the system to update the task board when changes occur to the cards.</td>
</tr>
<tr>
<td>Update Board on Lane Change</td>
<td>Visual Task Board Lane</td>
<td>Triggers the system to update the task board when changes occur to the lanes.</td>
</tr>
<tr>
<td>update board on lane change</td>
<td>Board Member</td>
<td>Triggers the system to update the task board when changes occur to the board members.</td>
</tr>
<tr>
<td>VTB Previous Additional Assignees</td>
<td>Private Task</td>
<td>Triggers the system to send an email notification to users when they are designated as additional assignees for cards.</td>
</tr>
</tbody>
</table>

### Notifications installed with Visual Task Boards

Visual Task Boards add the following notifications.

<table>
<thead>
<tr>
<th>Notification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTB additional assignee notification</td>
<td>Sends an email notification to users when they are designated as additional assignees for tasks.</td>
</tr>
<tr>
<td>VTB assignee notification</td>
<td>Sends an email notification to users when they are designated as the primary assignees for personal tasks.</td>
</tr>
<tr>
<td>VTB lane change notification</td>
<td>Sends an email notification to a task's assignees when the task changes lanes.</td>
</tr>
</tbody>
</table>

### Visual Task Board use

Any user can create, view, and edit freeform, flexible, and guided visual task boards, and add users to these boards as members.

You can create a freeform board and add cards from multiple task tables, or create a flexible or guided board from an existing list of records.

#### Create a freeform board

You can create new freeform task boards for tracking any kind of task or project.

Role required: none

1. Navigate to **Self-Service > Visual Task Boards.**
2. Under **Boards you own**, click **Create Freeform Board**.
   A new task board opens with the default freeform lanes, **To Do**, **Doing**, and **Done**.
3. Click the board name and enter a new name.

After the board is created, board members can:

- Add and modify lanes
- Create and modify task cards
- Add and remove members
- Configure the look and feel

### Create a flexible or guided board

You can create a flexible or guided task board for any table that extends Task, such as Incident or Change.

Role required: none

1. Navigate to the list for any table that extends Task, such as **Incident > All**.
2. Optional: **Create a filter** to show only the records you want to work with.
   For example, you might filter out incidents that are not assigned to you.
3. Right-click a column header and select **Show Visual Task Board**.
   The type of board that is created and the lanes that appear on the board depend on which type of column you select. If you select a reference or choice column, the board is a **guided board**. Each lane in a guided board represents one possible value for that column and the cards appear in the appropriate lane. For example, if you create a guided task board for incidents using the **State** column, the board shows one lane for each incident state, such as **New** or **Closed**. If you select a column that is not a reference or choice, the board is a **flexible board** and the board shows the default lanes, **To Do**, **Doing**, and **Done**, with all cards in the **To Do** lane.
4. Optional: Click the board name and enter a new name.

   **Note**: For guided boards created from choice fields, such as **State**, the system creates a lane for each possible choice value. For guided boards created from reference fields, the system creates a lane only for each value in use by a task card. Lanes are added as needed if the reference field values change.

After the board is created, board members can:

- Add and modify lanes
- Create and modify task cards
- Add and remove members
- Configure the look and feel

### Delete a task board

You can delete any task board you own.

Role required: none

Deleting a board does not impact the underlying task records. However, you cannot recover a board after you delete it.

1. Navigate to **Self-Service > Visual Task Boards**.
2. Under **Boards you own**, point to a board and click **Delete Board**.
Add or modify a lane

When you create a new freeform or flexible task board, it includes the default lanes **To Do**, **Doing**, and **Done**. Any board member can change the names and add new lanes to accommodate the task workflow you want to track.

Role required: none

Unlike guided boards, the lanes on a freeform or flexible task board do not match possible field values.

**Note:** You cannot remove lanes from guided boards.

1. Open a task board.
2. To create a new lane, scroll past all the existing lanes and click **Add Lane**.
   This option is available for guided boards only if the lanes are based on a reference field, such as **Assigned to**.
   A new, empty lane appears to the right of the existing lanes.
3. Enter a title for the lane, then press the Enter key.
4. To change the order of lanes, click the left and right arrows at the bottom of the lanes.
5. To delete a lane, click **Delete Lane** at the bottom of the lane.
   **Delete Lane** appears only if the lane does not contain any task cards. If you want to delete a lane that contains task cards, drag the task cards to another lane or archive the cards first.

Task cards

A Visual Task Board task card can represent a personal task or a record on a table.

You can add task cards, edit task card details, add labels to cards, and access the underlying task record the card represents, depending on the board type.

**Note:** Each freeform board can display up to 300 cards by default. Flexible and guided boards can display up to 100 cards. When the number of tasks exceeds the maximum, a warning appears and the system determines which cards to show by most recent update time. Administrators can configure different maximum card values.

Freeform board tasks

You can add personal task cards directly to lanes on a freeform board. You can also add other kinds of task cards to a freeform board from a list or form.

**Add a personal task to a freeform board**

You can add cards directly to a freeform board lane. These cards are called personal tasks and represent records on the Private Task [vtb_task] table.

Role required: none

1. On a freeform board, click the add card icon (+) beside the lane title. Alternatively, click **Add Task** in a lane that contains at least one card.
   A new task card appears in the lane.
2. Enter a short description for the new task card.
3. To save the card, click another area of the board or press **Enter**.

**Add a task to a freeform board from a form**

You can add a task card to a freeform board from any table that extends Task. These cards represent records on whichever table you add them from.
Role required: none
You can add cards from multiple task tables, such as Incident or Problem, to the same freeform board.

1. Open a task record, such as an incident.
2. Click the menu icon ( ) or right-click the form header to show the form context menu.
3. Select Add to Visual Task Board.
   A pop-up window appears with a list of freeform task boards.

4. Click the name of the board you want to add the card to.
   A confirmation message appears.
5. To view the board, click the board name in the confirmation message.
   The task appears in the first lane by default.

Add a task to a freeform board from a list
You can add a task card to a freeform board from any table that extends Task. These cards represent records on whichever table you add them from.

Role required: none
You can add cards from multiple task tables, such as Incident or Problem, to the same freeform board.

1. Navigate to a list of task records, such as Incident > Open.
2. Perform either of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add one task</td>
<td>Right-click a record and select Add to Visual Task Board</td>
</tr>
<tr>
<td>Add multiple tasks</td>
<td>Select the check box beside each record you want to add and select Add to Visual Task Board from the actions choice list</td>
</tr>
</tbody>
</table>

A pop-up window appears with a list of freeform task boards.
3. Click the name of the board you want to add the card to.
   A confirmation message appears.

4. To view the board, click the board name in the confirmation message.
   The task appears in the first lane by default.

Add a task to a flexible or guided board

You can add cards directly to a flexible or guided board lane. These cards represent records on whichever table the board is associated with.

Role required: none

1. On a flexible or guided board, click the add card icon (+) beside the lane title. Alternatively, click Add Task in a lane that contains at least one card.
   A pop-up window appears with a new record. Fields are populated based on any filter conditions for the board and, for guided boards, the lane you add the card to.
   Consider the example of a guided board that shows incidents with the Software category, with a lane for each assignee. If you add a card to the ITIL User lane, the new incident record Category and Assigned to fields are automatically set to Software and ITIL User.

2. Complete the form and click Submit.
   A new task card appears in the lane.

Edit card details

When you click a card on a Visual Task Board, a pop-up window appears with additional details.

Role required: none

You can edit certain values from the card details. Updating these values also updates the underlying task record.

1. Click a card.
   The details for that card appear in a pop-up window.

2. Alternatively, you can click an entry in the activity stream to view the details for the associated card.

3. From the card details you can:
   - Click the card title to rename the card.
   - Modify the short description.
   - Add comments or work notes in the activity stream.
   - Edit the assignees.
- Click one or more of the available labels to add a label to the card.
- Add or remove file attachments.
- Create a checklist.
- Archive the card (freeform boards only).
- Move the card to a different board (freeform boards only).

Label a task card

Labels help categorize tasks and visually distinguish them on the task board.

Enable labels for the board.
Role required: none

You can filter the visible tasks to show only those with certain labels. Labels appear on cards as colored dots.

![Image of task card labels](image)

Figure 599: Task card labels

You can add one or more labels to a task card. Do one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add a label to a card from the quick panel</strong></td>
<td>1. Drag a label from the quick panel directly to a task card.</td>
</tr>
<tr>
<td></td>
<td>2. Drop the label in the drop zone.</td>
</tr>
<tr>
<td><strong>Add a label to a card from the card details</strong></td>
<td>1. Open the card details for a task.</td>
</tr>
<tr>
<td></td>
<td>2. Click the details tab (i).</td>
</tr>
<tr>
<td></td>
<td>3. Under <strong>Labels</strong>, select one or more labels.</td>
</tr>
<tr>
<td></td>
<td>Point to a label to display its name.</td>
</tr>
<tr>
<td></td>
<td>4. Close the card detail pop-up window.</td>
</tr>
</tbody>
</table>

You can use configuration options to *rename* or *disable* labels or to *hide* labels altogether.

**Assign a task to a user**

Assignees represent ownership of task cards and the associated records. Each task card can have one primary assignee and multiple additional assignees.

You can filter the visible tasks on a board to show only tasks assigned to a certain user. Assignees appear on cards as avatars.
When a card changes lanes, all the assignees receive an email notification.

For flexible and guided boards, the primary assignee corresponds to the underlying task's **Assigned to** field. For example, on a board created from the Incident table, if you add Joe Employee as the primary assignee for a card, the associated incident is assigned to Joe. Note that assignment rules on the associated table may restrict who you can make the primary assignee of a card. If you attempt to assign a task card to a user in a way that violates an assignment rule, a notification appears and the assignment does not take place. There is no restriction on who you can add as an additional assignee.

---

**Note:** Assignees are not automatically granted access to the board. Only members can access the board.

---

Do one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add an assignee from the quick panel</strong></td>
<td>1. Drag a user's avatar from the quick panel directly to the task card.</td>
</tr>
<tr>
<td></td>
<td>2. Drop the avatar in the <strong>Primary</strong> or <strong>Additional</strong> drop zone.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Add an assignee from the card details       | 1. Open the card details for a task.  
  2. Click the assignees tab (👤).  
  3. Click **Add Primary Assignee** or **Add Additional Assignee**. The **Add Primary Assignee** option is not available if there is already a primary assignee.  
  4. Select a user.  
  5. Close the card detail pop-up window. |

The system sends an email notification to inform the new assignee of the change.

Visual Task Board checklists

Checklists help you track the progress of tasks on your Visual Task Board.

Checklists are available as a tab in the card details.
Figure 601: Sample checklist

You can add, remove, or rearrange checklist items. You can also save a checklist as a template for future use on other cards. Any user can create or edit a checklist. When the compact card view is disabled, the task card displays a progress bar to show how many items are complete.
You can also use checklists with task records in the standard interface. For more information, see Checklists on page 130.

Create a new checklist for a card
You can create a unique checklist for each Visual Task Board card.

Role required: none

1. Open the card details for a task.

2. Click the checklist tab (✓).

3. Click Add Item.

4. Enter text for the checklist item.

5. Press the Enter key to add the checklist item.

6. Create as many additional checklist items as desired.

7. Optional: To delete a checklist item, click the minus (-) icon.

8. Optional: To reorder checklist items, click the drag icon (_drag) and drag a checklist item to a different position in the list.

9. Optional: Save the checklist as a template for easy reuse.

   a) Click the more icon (•) beside the Checklist formatter.
b) Select **Save as Template**.
A dialog box appears.

c) Enter a descriptive template name to identify the checklist.
When a user creates a checklist from a template, all templates are listed in alphabetical order and there is no way to filter which templates appear. To provide a better user experience, consider implementing a naming system for checklist templates. For example, include the name of the table or another identifier to clarify how the checklist should be used.

d) Click **Save**.

10. Optional: To create a task from a checklist item:
   a) Point to a checklist item and click the create task icon (_drag_icon_).
   b) Select the type of task you want to create.
   An administrator can configure which task types are available.
   The new task record appears in a pop-up window. The short description is automatically populated with the checklist item text.
   c) Fill in the form with additional detail as necessary.
   d) Click **Update**.

**Create a checklist from a checklist template**
In a Visual Task Board card, you can quickly create a checklist from a checklist template you previously created.

Create at least one checklist template.
Role required: none
A template saves time by creating checklist items automatically. You can add, edit, or remove checklist items without impacting the template.

1. Open a task card that does not already contain a checklist.
2. Click the checklist tab (checkmark).
3. Click the more icon (arrow).
   The more icon does not display unless you have added an item to the checklist, or if you have a pre-existing checklist template.
4. Under **Copy items from**, select a template.
   The checklist items appear automatically.

**Delete a checklist from a card**
You can remove a checklist from a Visual Task Board card.
Role required: none

1. Open a task card that contains a checklist.
2. Click the checklist tab (checkmark).
3. Click the more icon (arrow).
4. Select **Remove Checklist**.
A confirmation dialog box appears.

5. Click **Delete**.

Configure which types of tasks can be created from checklist items

A system property controls which types of tasks you can create from a Visual Task Board checklist item.

Role required: admin

You might want to add custom tables or remove tables you do not use.

1. Navigate to `sys_properties.list`.
2. Locate the `glide.ui.create_task_type_option_list` property.
3. Edit the **Value** to add or remove tables.

You can add any table that extends Task [task]. Use the table name, not the label. For example, you would enter `change_request`, not Change Request. Ensure the tables listed are separated by commas, with no spaces.

The property value populates the task type selection list users choose from when they create a task from a checklist item. Note that users can only see task types for which they have the appropriate access roles.

![Add Item](image)

Archive a card

Archiving a card is a non-destructive way to remove a card from a freeform board.

Role required: none

You can archive cards you are no longer working on to reduce visual clutter. Note that it is only possible to archive cards on a freeform board.

1. To archive a card, view the card details and click **Archive Card**.
2. To see a list of archived cards for the current board, open the board configuration menu (gear) in the top-right corner of the board and click **View Archived Tasks**.
3. From the **Archived Tasks** pop-up window, you can restore cards to the board.
Access a task record

You can open the form view of any task card. Viewing a task record on a form allows you to see all fields for that record, including fields that are not visible in the card details.

Role required: none

Do one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access the record from the task card</td>
<td>Click the record number on the task card.</td>
</tr>
<tr>
<td>Access the record from the card details</td>
<td>1. Open the card details for a task.</td>
</tr>
<tr>
<td></td>
<td>2. Click the record number in the header.</td>
</tr>
</tbody>
</table>

Move a card to a different lane

You can move a card from one lane to another.

Role required: none

When you move a card to a different lane on a guided board, the field on which the lanes are based is updated to reflect the new lane value.

1. Open a Visual Task Board.
2. Select a card and drag it to a different lane.
   All mandatory fields on a record must be filled in to move the associated card to a new lane. For example, assume you have a guided board based on the Incident [incident] table, with a lane for each State. The Close code and Close notes fields become mandatory when the State changes to Closed. If you attempt to move an incident card from the New lane to the Closed lane, a pop-up view of the form appears and prompts you to fill in the mandatory fields.

Move a card to a different board

You can move a card from one freeform board to another.

1. Open the card details.
2. In the details tab (i), click Move Card.
3. In the pop-up window that appears, select the board and lane to move the card to.
   The board selection list displays freeform boards only.

Add or remove a task board member

You can add and remove task board members using the members tab of the activity stream. You can also promote assignees to board members.

Role required: none

Only members of a task board can access the board. Any board member can add or remove other members.

1. On a Visual Task Board, open the activity stream.
2. Click the members tab.
3. Perform one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a member</td>
<td>1. In the Members section, click the plus icon.</td>
</tr>
<tr>
<td></td>
<td>2. Enter a user's name or select a user from the drop-down menu.</td>
</tr>
<tr>
<td>Remove a member</td>
<td>1. In the Members section, point to a user.</td>
</tr>
<tr>
<td></td>
<td>2. Click Remove.</td>
</tr>
<tr>
<td>Promote an assignee to a member</td>
<td>1. In the Assignees section, point to a user.</td>
</tr>
<tr>
<td></td>
<td>2. Click Make Member.</td>
</tr>
<tr>
<td>Promote all assignees to members</td>
<td>In the Assignees section, click the promote all icon ( ).</td>
</tr>
</tbody>
</table>

View board information

You can display key information about any task board.

Role required: none

To view information about a visual task board, click the board information icon ( ) by the board name.

A flyout appears and displays the following information about the board:

- Name
- Board type
- Filter conditions (flexible and guided boards only)
- Owner
- Number of lanes
- Number of cards
- URL
Toggle the list view

For flexible and guided boards, you can switch to a list view of the tasks on the board.

Role required: none
<table>
<thead>
<tr>
<th>Incident ID</th>
<th>Caller</th>
<th>Description</th>
<th>Category</th>
<th>Priority</th>
<th>State</th>
<th>Assignment group</th>
<th>Assigned to</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC00000002</td>
<td>Joe Employee</td>
<td>Unable to get to network file shares</td>
<td>Network</td>
<td>1 - Critical</td>
<td>Awaiting Problem</td>
<td>Network</td>
<td>Howard Johnson</td>
</tr>
<tr>
<td>INC00000003</td>
<td>Joe Employee</td>
<td>Wireless access is down in my area</td>
<td>Network</td>
<td>1 - Critical</td>
<td>Active</td>
<td>Network</td>
<td>Beth Angin</td>
</tr>
<tr>
<td>INC00000004</td>
<td>Joe Employee</td>
<td>Need access to sales DB for the West</td>
<td>Database</td>
<td>1 - Critical</td>
<td>Awaiting User Info</td>
<td>Database</td>
<td>David Loo</td>
</tr>
<tr>
<td>INC00000005</td>
<td>Fred Luddy</td>
<td>I can't launch my VPN client since the last software update</td>
<td>Software</td>
<td>1 - Critical</td>
<td>Active</td>
<td>Software</td>
<td>Don Goodlife</td>
</tr>
<tr>
<td>INC00000006</td>
<td>Bow Ruggel</td>
<td>Rain is leaking on main DNS server</td>
<td>Hardware</td>
<td>1 - Critical</td>
<td>Active</td>
<td>Hardware</td>
<td>ITIL User</td>
</tr>
<tr>
<td>INC00000007</td>
<td>Joe Employee</td>
<td>How do I create a sub-folder</td>
<td>Inquiry/Help</td>
<td>1 - Critical</td>
<td>Awaiting User Info</td>
<td>Service Desk</td>
<td>Fred Luddy</td>
</tr>
<tr>
<td>INC00000008</td>
<td>Taylor Yeats</td>
<td>Sales forecast spreadsheet is READ ONLY</td>
<td></td>
<td>1 - Critical</td>
<td>Active</td>
<td>ITIL User</td>
<td></td>
</tr>
<tr>
<td>INC00000009</td>
<td>Fred Luddy</td>
<td>Can't launch 64-bit Windows 7 virtual machine</td>
<td>Software</td>
<td>2 - High</td>
<td>Active</td>
<td></td>
<td>Bud Richman</td>
</tr>
<tr>
<td>INC00000010</td>
<td></td>
<td>I need a replacement iPhone, please</td>
<td>Request</td>
<td>5 - Planning</td>
<td>Active</td>
<td>ITIL User</td>
<td></td>
</tr>
<tr>
<td>INC00000011</td>
<td>Don Goodlife</td>
<td>I need more memory</td>
<td>Hardware</td>
<td>1 - Critical</td>
<td>Active</td>
<td>ITIL User</td>
<td></td>
</tr>
<tr>
<td>INC00000012</td>
<td>Fred Luddy</td>
<td>Please remove the latest hotfix from my PC</td>
<td>Software</td>
<td>2 - High</td>
<td>Active</td>
<td>ITIL User</td>
<td></td>
</tr>
<tr>
<td>INC00000013</td>
<td>Charlie Kitchen spoon</td>
<td>I can't get my weather report</td>
<td>Request</td>
<td>5 - Planning</td>
<td>Active</td>
<td>Service Desk</td>
<td>Don Goodlife</td>
</tr>
<tr>
<td>INC00000014</td>
<td>Joe Employee</td>
<td>When can we get rid of My? UI is killing us</td>
<td>Inquiry/Help</td>
<td>1 - Critical</td>
<td>Active</td>
<td></td>
<td>David Loo</td>
</tr>
<tr>
<td>INC00000015</td>
<td>Sam Scrakin</td>
<td>Request for a new service</td>
<td>Request</td>
<td>3 - Moderate</td>
<td>Active</td>
<td>Service Desk</td>
<td>Howard Johnson</td>
</tr>
<tr>
<td>INC00000016</td>
<td>Bud Richman</td>
<td>Trouble getting to Oregon mail server</td>
<td>Network</td>
<td>5 - Planning</td>
<td>New</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A list of records appears in place of lanes and cards.

2. Use the list like any other list.

   Note: You cannot use breadcrumbs.

3. Click Lanes by the board filter to switch back to the lane view of the tasks.

Visual Task Board SLA indicators

SLA (service level agreement) indicators alert board members if any cards on the board have breached SLAs.

When SLA indicators are enabled, a status bar appears on each card that has an associated SLA. If any cards have breached SLAs, an alert appears at the top of the board and a filter option appears in the quick panel. The indicator displays the elapsed percentage of the SLA. If there are multiple SLAs associated with a task card, the indicator displays the status of the SLA with the greatest elapsed percentage.
Figure 603: Visual Task Board with SLA indicators enabled

ALERT! One or more tasks on this board require immediate attention

New

EMAIL Server Down

SLA: 51%
Urgency: 1 - High
Priority: 1 - Critical
INC0000031
2h ago

Active

There seems to be some slowness or an out to SAP Materials Management

The SAP Materials Management Handbook

SLA: 21%
Assigned to: System Administrator

Awaiting Problem

please revert caching perfomance hotfix

SLA: 111%
Assigned to: Boris Catino
Urgency: 1 - High
Priority: 1 - Critical
Any board member can enable SLA indicators from the board configuration menu. The setting applies to the current user only.

Share a Visual Task Board in a Connect conversation

You can share a Visual Task Board in a Connect Chat or Connect Support conversation.

Role required: none

1. Navigate to **Self-Service > Visual Task Boards**.
2. Drag and drop a task board to a Connect mini window.
   
   A link to the task board appears in the conversation. The task board is also listed in the conversation tools, which are visible in the Connect workspace. Note that only conversation members who are members of the board can access it. If you share a task board in a record conversation, it appears as a URL in the record activity stream.

Create a Connect conversation from a Visual Task Board

Create a Connect Chat conversation from a Visual Task Board to collaborate with board members and keep track of activity as it happens.

Role required: none

When you create a conversation from a task board, all the board members become members of the conversation. Adding or removing users from the board members list automatically adds or removes them from the conversation. However, adding or removing users from the conversation does not add or remove them from the board members list.

1. Navigate to **Self-Service > Visual Task Boards**.
2. Open a task board.
3. In the board header, click the connect with board members icon ($$$\text{_BOARD}$$$).
   
   The system automatically opens a record conversation for the board. Each board member becomes a member of the conversation.

Configure the task board

The board configuration menu provides several options that affect the look and feel of the board.

Role required: none

1. Click the menu icon ($$$\text{_CONFIG}$$$) in the top-right corner of the board.
   
   The **Quick Panel** and **Configure Cards** sections provide several options that affect the look and feel of the board. These options are enabled when the switch appears green and is toggled to the right.

2. Select a background color from the palette.
The background color surrounds the lane area of a board. It also appears as the color for the board on the My Task Boards page. Changes to the background color apply to all users.

3. Configure quick panel options, as necessary.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Panel</td>
<td>When enabled, the quick panel is visible. This setting applies to the current user only.</td>
</tr>
<tr>
<td>User Names</td>
<td>When enabled, user names appear by user avatars in the quick panel. This setting applies to the current user only.</td>
</tr>
<tr>
<td>Show Labels</td>
<td>When enabled, labels appear on cards and in the quick panel. This setting applies to all users.</td>
</tr>
<tr>
<td>Label Names</td>
<td>When enabled, the label name appears beside the colored icon for that label in the quick panel. This setting applies to the current user only and is available only when Show Labels is enabled.</td>
</tr>
</tbody>
</table>

When all the options are enabled, a quick panel looks like this:

![Quick panel](image)

4. Configure card options, as necessary.

Changes to the cards apply to the current user only.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact Cards</td>
<td>When enabled, the lane width is decreased and card information and thumbnails do not appear on the card.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Show Card Info</td>
<td>When enabled, card information appears on each card. Card information typically includes details like the task state, date opened, and last user to update it. This setting is available only when <strong>Compact Cards</strong> is disabled.</td>
</tr>
<tr>
<td>Show Card Thumbnail</td>
<td>When enabled, cards display the first image attached to that card as a thumbnail. This setting is available only when <strong>Compact Cards</strong> is disabled.</td>
</tr>
<tr>
<td>Show SLAs</td>
<td>When enabled, SLA status indicators appear on cards and alerts appear when one or more card has a breached SLA. SLA status indicators are hidden when <strong>Compact Cards</strong> is enabled. This setting is available only when there are SLAs associated with the tasks on the board.</td>
</tr>
</tbody>
</table>

5. Show or hide lanes by selecting or clearing the check box for each lane, listed under **Configure Lanes**.

**Configure Visual Task Board labels**

Labels help categorize tasks and visually distinguish them on the task board. You can disable or rename labels using the labels tab of the activity stream.

Role required: none

You can use up to five labels per board. Any board member can edit the labels on a board.

1. Open the activity stream.
2. Click the labels tab.
3. Toggle the **Show Labels** switch to the enabled position if it is not already enabled.
4. To change a label name, click the label text and enter a new value.
5. To disable or enable a specific label, toggle the switch by the label.

When a label is disabled, members cannot add that label to cards on the board. Cards that are already labeled are not affected.
Configure the card limit for freeform boards

Freeform boards can display up to 300 cards by default. You can change the default card limit by adding a property.

Role required: admin

1. **Add a new system property** with the following field values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.vtb.freeform_max</td>
</tr>
<tr>
<td>Type</td>
<td>integer</td>
</tr>
</tbody>
</table>

2. Set the **Value** to the maximum number of cards allowed for each freeform board.

Configure the card limit for flexible and guided boards

Flexible and guided boards can display up to 100 cards by default. You can change the default card limit by adding a property.

Role required: admin

1. **Add a new system property** with the following field values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.vtb.board_max</td>
</tr>
<tr>
<td>Type</td>
<td>integer</td>
</tr>
</tbody>
</table>

2. Set the **Value** to the maximum number of cards allowed for each flexible or guided board.

Connect

ServiceNow Connect is a real-time messaging platform that connects you to your coworkers, bypassing email and static documents.

Connect integrates with other features within the system, such as Visual Task Boards, HR Service Management, and Customer Service Management. The Connect interface overlays the standard interface, which allows users to participate in conversations while they work.

Connect contains the following features:

- Connect Chat: Enables users to chat with individuals and groups, quickly share files, and collaborate on any record by connecting with the right people instantly.
- Connect Support: Enables support agents to provide real-time assistance to end users, using queues. Requires the Connect Support plugin.

UI16 or UI15 is required to use Connect.

**Note:**

- Connect does not replace legacy chat but offers some of the same functionality. The features should not be used concurrently.
• Connect is called Collaboration in versions prior to Geneva.

---

**Supported browsers for Connect**

The system supports Connect Chat and Connect Support on most modern browsers.

- The latest public release of Firefox or Firefox ESR
- The latest public release of Chrome
- Safari version 6.1 and later
- Internet Explorer version 10 and later
  - Edge mode is supported.
  - Compatibility mode is not supported.
  - Setting Security Mode to High (via the Internet Options > Security tab) is not supported.
  - Internet Explorer 11 is susceptible to memory leaks, which may impact performance, especially in Windows 7.

**Activate Connect**

Connect is active by default on new instances. For upgraded instances, you can activate the Connect plugin (com.glide.connect) if you have the admin role.

Role required: admin

If you used Collaboration in the Fuji release, the Connect plugin is activated automatically.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

**Installed with Connect**

Several types of components are installed with Connect.

**Tables installed with Connect**

Connect adds the following tables.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborator</td>
<td>Stores a record for each conversation, for each conversation member.</td>
</tr>
<tr>
<td>[collaborator]</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> The Collaborator table is left over from the Collaborator plugin in Fuji. It is not used in Connect or Live Feed.</td>
<td></td>
</tr>
<tr>
<td>Connect Action</td>
<td>Stores a record for each Connect action. This table is also used in Connect Support.</td>
</tr>
<tr>
<td>[connect_action]</td>
<td></td>
</tr>
</tbody>
</table>

Properties installed with Connect

Connect adds the following system properties.

**Note:** All of the properties are located on the System Property [sys_properties] table. To open the System Property [sys_properties] table, enter `sys_properties.list` in the navigation filter.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collaboration.email_interval</td>
<td>Sets the number of minutes the system waits before sending a Connect notification email to an inactive user.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> 3</td>
</tr>
<tr>
<td></td>
<td>• <strong>More information:</strong> Configure the email notification interval</td>
</tr>
<tr>
<td>collaboration.frameset</td>
<td>Determines whether the Connect overlay is visible (enabled). This property also impacts Connect Support.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> true</td>
</tr>
<tr>
<td></td>
<td>• <strong>More information:</strong> Disable the Connect overlay on page 2283</td>
</tr>
<tr>
<td>collaboration.polling_interval</td>
<td>Sets the number of seconds the system waits between polling for new Connect messages. This property also impacts Connect Support.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> 10</td>
</tr>
</tbody>
</table>
### Name: glide.connect.enabled

**Description:** Hides the Create or Join Chat Room related link, which appears on the Incident form when legacy chat is enabled. Legacy chat and Connect Chat should not be used concurrently and this property should not be modified.

- **Type:** true | false
- **Default value:** true

### Name: glide.live_feed.task_header_button

**Description:** Determines whether the show live feed icon (_follow_ button) are available in the form header of tables that have the live_feed=true dictionary attribute. The available choices are both, live_feed, collaboration, and none.

- **Type:** choice list
- **Default value:** collaboration
- **More information:** Configure record conversations

---

## Business rules installed with Connect

Connect adds the following business rules.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add message to conversations</td>
<td>Visual Task Board Card [vtb_card]</td>
<td>Generates system messages for conversations created from Visual Task Boards. The messages provide updates when cards are added to the board or moved between lanes.</td>
</tr>
<tr>
<td>Broadcast Chat APNS</td>
<td>Live Feed Message [live_message]</td>
<td>Generates push notifications when new messages are received in a conversation.</td>
</tr>
<tr>
<td>Broadcast Chat APNS</td>
<td>Live Mention [live_mention]</td>
<td>Generates push notifications when a user is mentioned in a conversation.</td>
</tr>
<tr>
<td>Sync Board Members With Group Members</td>
<td>Live Group Member [live_group_member]</td>
<td>Adds Visual Task Board members as conversation members when you create a conversation from the board.</td>
</tr>
</tbody>
</table>

## Notifications installed with Connect

Connect adds the following notifications.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration Offline Group Message(s)</td>
<td>Sends an email notification to offline users when there are new messages in group chats they are members of. Contains a link to open the conversation and a summary of the messages.</td>
</tr>
<tr>
<td>Collaboration Offline Message Bundle</td>
<td>Sends an email notification to offline users when there are new messages in direct chats. Contains a link to open the conversation and a summary of the messages.</td>
</tr>
<tr>
<td>ConnectMessagePushNotification</td>
<td>Sends push notifications to users according to their notification preferences. This notification also impacts Connect Support.</td>
</tr>
</tbody>
</table>

### Activate Connect Support

You can activate the Connect Support plugin (com.glide.connect.support) if you have the admin role. This plugin includes demo data. If you are currently using the legacy chat feature to provide help desk support, close any open help desk chats before activating Connect Support. Legacy chat and Connect Support cannot be used concurrently. Starting with Geneva Patch 8, when you activate Connect Support, the system automatically sets the state of all Chat Queue Entry [chat_queue_entry] records to Closed Complete. This ends any open help desk chats.

**Role required:** admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

### Installed with Connect Support

Several types of components are installed with Connect Support.

**Note:** Connect Support also utilizes many of the components installed with Connect.

### Tables installed with Connect Support

Connect Support adds the following tables for system use.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chat Queue Entry Transfer [chat_queue_entry_transfer]</td>
<td>Stores a record each time an agent transfers a support conversation to another agent.</td>
</tr>
</tbody>
</table>
Properties installed with Connect Support

Connect Support adds the following system properties.

**Note:** All of the properties are located on the System Property [sys_properties] table. To open the System Property [sys_properties] table, enter `sys_properties.list` in the navigation filter.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type:</th>
<th>Default value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect.support.conversation_limit</td>
<td>Determines how many support conversations an individual agent can have at one time. When the value is set to -1, an agent can participate in an unlimited number of conversations.</td>
<td>integer</td>
<td>-1</td>
</tr>
<tr>
<td>connect.support.show_agent_avatar</td>
<td>Determines whether an agent's avatar is shown in a support conversation (enabled). When the property is disabled, users see the agent's name only.</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>connect.support.user.closed.conversation_limit</td>
<td>Determines how many closed conversations appear in a user's support conversation history. When the value is set to 0, all previous conversations appear in the history.</td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td>glide.connect.support.enabled</td>
<td>Disables or enables Connect Support. When the property is enabled, all help desk conversations are opened in Connect Support, rather than legacy chat. Additionally, the support tab appears in the Connect sidebar. This property is for system use only and should not be modified.</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>connect.support.idle.count_down</td>
<td>Number of seconds to count down from before marking end-user as having left their support session.</td>
<td>integer</td>
<td>60</td>
</tr>
</tbody>
</table>


Available starting with Geneva Patch 2.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect.support.idle.delay</td>
<td>Number of seconds to wait (without user interaction), before presenting end-users with an idle countdown timer.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> 120</td>
</tr>
</tbody>
</table>

Business rules installed with Connect Support

Connect Support adds the following business rules.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNC - Chat Queue Average Wait Time</td>
<td>Chat Queue Entry [chat_queue_entry]</td>
<td>Populates the <strong>Average wait time</strong> field on the Chat Queue table with a calculated value when a support conversation is accepted. The calculation is based on the last 20 support conversations for the queue.</td>
</tr>
<tr>
<td>SNC - Chat Queue Entry Duration</td>
<td>Chat Queue Entry [chat_queue_entry]</td>
<td>Populates the <strong>Duration</strong> field on the Chat Queue Entry table with a calculated value, based on the amount of time the support conversation was open.</td>
</tr>
</tbody>
</table>

Connect interface

Users can work in a compact view of Connect, which overlays the standard user interface, or in a full-screen workspace.

Connect Chat and Connect Support share the same interface, so support agents can keep track of all their conversations in one place.

Connect overlay

The Connect overlay appears over the standard user interface. It consists of the Connect sidebar and any Connect mini windows that are open.
Note: An administrator can disable the Connect overlay so users can only use the Connect workspace, a full-screen interface with additional Connect tools.

Connect sidebar

The Connect sidebar is the primary interface for Connect Chat and Connect Support. It lists your conversations and provides access to create new conversations.

The sidebar is collapsed by default. Click the toggle Connect sidebar icon ( ) in the banner frame to expand or collapse the sidebar, which appears on the right edge of the interface.

Connect Chat

The chat view of the sidebar is available to all users. If Connect Support is enabled, users must click the chat tab of the sidebar, which is represented by a speech bubble icon. If Connect Support is not enabled, there are no tabs in the sidebar and the chat view displays by default.

The chat view of the sidebar lists all your current Connect Chat conversations. It also contains a filter tool and a button to create new conversations.
If you have unread messages in a conversation, a number appears by the conversation in the sidebar. Point to a conversation and click the x icon to remove the conversation from the sidebar.

**Connect Support**

The support view of the sidebar is available when Connect Support is activated. The support view is visible only to users who are agents for at least one support queue. Users must click the support tab of the sidebar, which is represented by a headset icon.

The support view displays **Queues** to which you belong. It also displays your open support conversations under **Cases**.
If you have unread messages in a conversation, a number appears by the conversation in the sidebar.

Figure 606: Support view of the Connect sidebar

Connect mini windows

When you open a Connect Chat or Connect Support conversation in the Connect overlay, it opens in a Connect mini window. Each mini window contains a header, a conversation area, and a message field. Several icons appear in the mini window header and by the message field.

Note: The icons you see for each conversation depend on the conversation type and other conditions.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>👤+</td>
<td>Add user</td>
<td>Add a user to the conversation. This icon is visible for group and record conversations only.</td>
</tr>
<tr>
<td>Icon</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>View document</td>
<td>View the record associated with the conversation. This icon is visible for record and support conversations only.</td>
</tr>
<tr>
<td></td>
<td>New window</td>
<td>Open the conversation in the Connect workspace, a full-screen window with additional Connect tools.</td>
</tr>
<tr>
<td></td>
<td>Collapse/Expand conversation</td>
<td>Hide or show the conversation. Alternatively, click anywhere in the mini window header to the left of the icons.</td>
</tr>
<tr>
<td></td>
<td>Close window</td>
<td>Close the mini window. The conversation remains in the Connect sidebar and the history is preserved when you reopen the conversation.</td>
</tr>
<tr>
<td></td>
<td>Connect actions</td>
<td>Open the Connect actions menu, which contains options like Transfer and Create Incident. This icon is visible only when Connect actions are available for the conversation. By default, this icon is visible for support conversations only. Administrators can add options to this menu by creating new Connect actions.</td>
</tr>
<tr>
<td></td>
<td>Message type</td>
<td>Choose whether messages are added to the associated record as comments or work notes. This icon is visible only for record and support conversations.</td>
</tr>
<tr>
<td></td>
<td>Attach file</td>
<td>Attach and send a file. Select one or more files and click Open. You can also add an attachment by dragging and dropping it directly in the conversation.</td>
</tr>
</tbody>
</table>

**Connect workspace**

The Connect workspace is a full-screen view of all your Connect Chat and Connect Support conversations in one place. It provides additional tools to help keep track of important information in conversations.

To open the Connect workspace, navigate to **Collaborate > Connect Chat** or click the new window icon ( ) in a Connect mini window. If you do not have any recent conversations, a screen appears with helpful information about Connect.
The Connect workspace is composed of the following elements.

Table 552: Connect workspace elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sidebar</td>
<td>Provides access to conversations. The Connect sidebar behaves the same way in the workspace as it does in the Connect overlay. The only difference is that the sidebar appears on the left edge of the Connect workspace. For more information about using the sidebar, see <a href="#">Connect sidebar</a>.</td>
</tr>
<tr>
<td>Conversation pane</td>
<td>Displays the currently selected conversation.</td>
</tr>
<tr>
<td>Conversation tools</td>
<td>Provides quick access to key information, conversation members, attachments, and notification preferences for the currently selected conversation. Some of the conversation tools vary depending on the type of conversation.</td>
</tr>
</tbody>
</table>
Figure 607: Connect workspace
Conversation pane

The conversation pane of the Connect workspace displays the conversation header and an expanded version of the mini window.

The header displays basic conversation details like the avatar and name. You can edit these details for group or record conversations.

Below the header, messages appear in chronological order. You can enter messages in the text entry field at the bottom of the window.

Connect conversation tools

The conversation tools area in the Connect workspace contains several tabs, each represented by an icon.

Click an icon to open the tab. Click the same icon again to hide the tab. The tabs you see for each conversation depend on the conversation type and other conditions.
<table>
<thead>
<tr>
<th>Tab icon</th>
<th>Tab name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Info](image) | Info | Contains the following sections, each of which appears only if it contains information.  
  - **Record**: Lists details about the record the current conversation follows, such as record number, state, assignee, and short description. The fields displayed in this section vary by the type of record. This section is available only for record conversations.  
  - **Related Records**: Lists task records that have been referenced in the conversation, such as incidents, problems, or changes. When you send a record number as a message or drag and drop a record in a conversation, it appears as a link. The list displays the short description for each task. Click a task to open the record in a new browser tab. Note that only conversation members who have rights to view the tasks can access them.  
  - **Links**: Lists URLs that have been referenced in the conversation. Click a link to open the destination page in a new browser tab.  
  - **Visual Task Boards**: Lists Visual Task Boards that have been referenced in the conversation. Click a Visual Task Board name to open the board in a new browser tab. |

This tab is not available for support conversations.
<table>
<thead>
<tr>
<th>Tab icon</th>
<th>Tab name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Tab icon" /></td>
<td>Record</td>
<td>Displays a compact form view of a record created from the current conversation, such as an incident. If more than one record has been created from the conversation, there is a separate record tab for each one. These tabs are available only for support conversations.</td>
</tr>
<tr>
<td><img src="image2" alt="Tab icon" /></td>
<td>Knowledge Base</td>
<td>Displays a compact view of the knowledge homepage. For more information, see <a href="#">Share knowledge in a Connect Support conversation</a> on page 2292.</td>
</tr>
<tr>
<td><img src="image3" alt="Tab icon" /></td>
<td>Members</td>
<td>Lists all members of the current conversation. You can add or remove conversation members for group and record conversations.</td>
</tr>
<tr>
<td><img src="image4" alt="Tab icon" /></td>
<td>Attachments</td>
<td>Lists all attachments in the conversation. Click an attachment to open it. Click <strong>Add Attachment</strong> to upload an attachment.</td>
</tr>
<tr>
<td><img src="image5" alt="Tab icon" /></td>
<td>Notification Preferences</td>
<td>Contains settings to control which notifications you receive for the current conversation. For more information, see <a href="#">Edit which notifications you receive for a conversation</a> on page 2278.</td>
</tr>
</tbody>
</table>

### Enable or disable Connect notifications globally

You can edit your Connect notification settings globally to enable or disable mobile, desktop, or email notifications for all your Connect Chat and Connect Support conversations.

**Role required:** none

By default, Connect mobile and email notifications are enabled globally. You must enable desktop notifications.

**Note:** You receive mobile notifications only if you have installed the ServiceNow mobile app.

1. In the banner frame, click the gear icon (⚙️) to open the system settings.
2. In the system settings window, click the **Notifications** tab.
3. Enable or disable Connect mobile, desktop, or email notifications.
Edit which notifications you receive for a specific conversation.

**Connect notification browser support**

The following browsers support Connect chat notifications.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Enable notification steps</th>
</tr>
</thead>
</table>
| Chrome                         | 1. From a Chrome window, click the Chrome menu in the upper right corner.  
|                                | 2. Go to **Settings > Show advanced settings... > Privacy > Content settings**.  
|                                | 3. In the Content settings window, from the notifications section, select your notification settings. |
| Mozilla Firefox (starting with version 44) | Manage browser notification settings using the information icon next to the URL. |
| Safari (starting with OS X Mavericks) | 1. From a Safari window, open **Preferences**.  
|                                | 2. Select the notifications tab.  
|                                | 3. Select **Allow** or **Deny** to control which websites have access to notifications. |
| Microsoft Edge                 | Notifications are enabled by default on Windows 10 PCs and tablets starting with EdgeHTML 14. For more information, see [https://blogs.windows.com/msedgedev/2016/05/16/web-notifications-microsoft-edge/#pVkwubg7uvROayyl97](https://blogs.windows.com/msedgedev/2016/05/16/web-notifications-microsoft-edge/#pVkwubg7uvROayyl97). |

**Note:** Internet Explorer does not support Connect notifications out of box.

**Edit which notifications you receive for a conversation**

You can control which types of Connect notifications you receive for each conversation and when you receive them.

Ensure Connect notifications are **enabled globally**.

Role required: **none**

Connect can send mobile, desktop, and email notifications to inform you of messages and other activity, when you are not actively viewing Connect. You can edit your notification preferences in each conversation to control which notifications you receive for the conversation and under what circumstances. For example, if you are a member of a large group conversation, you might want to receive mobile and email notifications for all activity, and desktop notifications only when someone mentions you. You might also want to disable a certain type of notification entirely for a conversation.
Audio notifications are unavailable for Connect in Geneva.

By default, you receive mobile and email notifications for all activity in every conversation you belong to. You must enable desktop notifications manually. Not all browsers allow desktop notifications. For more information, see Connect notification browser support on page 2278.

**Note:** You receive mobile notifications only if you have installed the ServiceNow mobile app.

1. Open a conversation in the Connect workspace.
2. In the conversation tools to the right of the conversation, click the notification settings tab ( ).
3. Edit the following settings according to your preferences.

**Note:** If a notification type is disabled globally, a link to enable that type appears. You must click the link before you can edit conversation-specific notification settings.

![Notification type disabled globally](image)

**Figure 608: Notification type disabled globally**

<table>
<thead>
<tr>
<th>Mobile Notification Settings</th>
<th>Choose to receive push notifications:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• For all activity</td>
</tr>
<tr>
<td></td>
<td>• Only when @ mentioned (this option is not available for direct conversations)</td>
</tr>
<tr>
<td></td>
<td>• Never</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Desktop Notification Settings</th>
<th>Choose to receive desktop notifications:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• For all activity</td>
</tr>
<tr>
<td></td>
<td>• Only when @ mentioned (this option is not available for direct conversations)</td>
</tr>
<tr>
<td></td>
<td>• Never</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Email Settings</th>
<th>Choose to receive email notifications:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• For all activity</td>
</tr>
<tr>
<td></td>
<td>• Only when @ mentioned (this option is not available for direct conversations)</td>
</tr>
<tr>
<td></td>
<td>• Never</td>
</tr>
</tbody>
</table>

| System Message Notifications | Enable or disable system message notifications, which are automatically generated by events like conversation membership changes and Visual |
Upload a profile picture

You can upload a profile picture in your Live Feed profile to use as an avatar in Connect Chat and Connect Support conversations.

Role required: none

Users who do not have profile pictures are represented by an avatar with their initials. Administrators can upload a profile picture in a user record which displays if there is no Live Feed photo.

1. Navigate to Self-Service > Live Feed.
2. Click the tile with your picture or initials and title.

3. Use one of the following actions to add your photo.
   - Locate the photo file you want to use and drag it over the existing photo.
   - Point to the existing picture (or tap the photo in the smartphone or tablet interface) to display the Upload a picture link. Click the link, navigate to the location of the photo you want to use, and click Open.
Drag and drop a file in a Connect conversation

Drag and drop functionality in Connect Chat and Connect Support provides an easy way to share external attachments and links or items from within your instance.

Role required: none

You can drag and drop several things from within an instance, including items from the application navigator, records or breadcrumbs from lists, and Visual Task Boards from the My Task Boards page.

Drag and drop an item in a Connect mini window.

To drag and drop a record from a list, drag the reference icon ( ), the record number, or a reference column value.

Share a Visual Task Board in a Connect conversation

You can share a Visual Task Board in a Connect Chat or Connect Support conversation.

Role required: none

1. Navigate to **Self-Service** > **Visual Task Boards**.
2. Drag and drop a task board to a Connect mini window.

A link to the task board appears in the conversation. The task board is also listed in the conversation tools, which are visible in the Connect workspace. Note that only conversation members who are members of the board can access it. If you share a task board in a *record conversation*, it appears as a URL in the record activity stream.

Mention a user in a Connect conversation

You can get someone's attention in a group conversation by mentioning them.

Role required: none

Users can set their notification preferences so they receive notifications only when they are mentioned. Mentioning a user creates a shortcut to view their basic information and send them a direct message.
1. Open a Connect group conversation.
2. In the message field, type the @ character. A suggestion menu appears with the names of the conversation members.
3. Select the person you want to mention and send the message.

Connect administration

Administrators can configure various performance settings and features that impact both Connect Chat and Connect Support.

**Note:** There are also administrative options specifically for Connect Chat or Connect Support. For more information, see Connect Chat administration on page 2288 and Connect Support administration on page 2296.

Configure the polling interval

The polling interval determines how frequently the system polls for new Connect messages.

Role required: admin

The default interval is 10 seconds. You can change this value. The shorter the polling interval, the more frequently the system checks for new messages and the greater the impact on performance.
Note: This setting impacts Connect Chat and Connect Support.

1. Navigate to `sys_properties.list`.
2. Locate the `collaboration.polling_interval` property.
3. Set the Value to a different number of seconds.

   Setting the polling interval to a value smaller than 2 is likely to tax the system too heavily, while a value greater than 10 is likely to result in a poor user experience.

Enable the Connect overlay

The Connect overlay is enabled by default and is integrated with the standard user interface. You can disable the Connect overlay.

Role required: admin

To disable the Connect overlay and remove the toggle Connect sidebar icon from the banner frame, set the `collaboration.frameset` property to `false`.

1. Navigate to `sys_properties.list`.
2. Locate the `collaboration.frameset` property.
3. Set the Value to `false`.

Administer Connect actions

You can create or modify Connect actions to provide custom functionality in Connect Chat or Connect Support conversations.

Role required: admin

The Connect action menu appears by the message entry field when one or more Connect actions are available for a conversation, based on defined conditions. When a user selects a Connect action, the system runs the script defined for that action.

1. Navigate to `connect_action.list` to access the Connect Action table.
   
   If Connect Support is enabled, you can also navigate to Collaborate > Administration > Actions.
2. Click New or open an existing Connect action.
3. Complete the Connect Action form, as appropriate.

Table 555: Connect Action form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>JavaScript condition statement that must return true for the action to be available in a conversation. For example, to show the action in Connect Support conversations only, enter <code>conversation.type === &quot;support&quot; or conversation.table === 'chat_queue_entry'</code>. For information about the <code>conversation</code> object, see the GlideConversation API reference documentation.</td>
</tr>
<tr>
<td>Hint</td>
<td>This field is not used.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Icon Class Name</td>
<td>This field is not used.</td>
</tr>
<tr>
<td>Order</td>
<td>Order of the action relative to other items in the Connect action menu. This field is available starting with Geneva Patch 6.</td>
</tr>
<tr>
<td>Script</td>
<td>Script to execute when the action is run. For example, to create a new incident based on the conversation, enter the following code:</td>
</tr>
</tbody>
</table>
|              |     response.newRecord("incident", { short_description: conversation.document.short_description || ",
caller_id: conversation.document.opened_by }); |
|              | For information about the response object, see the ConnectActionResponse API reference documentation.                                        |
| Shortcut     | Text that triggers the action when entered after the "/" character in a conversation.                                                        |
| Title        | Text that appears in the Connect action menu. The title for each Connect action should be unique.                                             |
Connect Chat

Connect Chat is a real-time messaging tool that enables users to chat with individuals and groups, quickly share files, and collaborate on any record by connecting with the right people instantly.

Connect Chat animates communication around records, Visual Task Boards, topics of interest, or groups of people.

Features include:

- Direct conversations between two users.
- Group conversations between three or more users.
• Conversations linked to records. Comments and work notes appear in conversations in real time and users can update the record directly from the conversation.
• Drag-and-drop sharing of links, files, and records.

UI16 or UI15 is required to use Connect Chat.
This video demonstrates how to use Connect Chat.
This video demonstrates how to use Connect Support.

Note:
• Connect Chat does not replace legacy chat but offers some of the same functionality. The features should not be used concurrently.

Connect Chat use

All users have access to Connect Chat, a convenient way to stay updated on all the people and documents you work with in a familiar chat interface.

You can start using Connect Chat right away with virtually no setup. Connect Chat supports a few different types of conversations so you can use it as a social tool or a quick way to follow specific records.

Start a direct or group conversation

You can start a conversation with one or more users in Connect Chat.

Role required: none

A conversation between two users is called a direct conversation. A conversation between three or more users is called a group conversation.

1. In the chat view of the Connect sidebar, click the plus icon (+) by the filter.
   A Connect mini window appears.
2. In the To field, enter the name of the user you want to chat with.
   Suggestions appear as you type.
3. Optional: Enter additional names to create a group conversation.
   You can add or remove group conversation members later. You cannot add members to a direct conversation after you create it.
4. Press the Enter key to shift focus to the message field.
5. Send a message to create the conversation.
   The conversation is added to the Connect sidebar for quick access.

Follow a record in Connect

You can follow a record in Connect Chat to track activity as it happens, in a familiar chat interface. When you follow a record, you become a member of a record conversation.

Role required: whichever role is required to access the record you want to follow in Connect

Any record conversation you create in Connect becomes a record feed in live feed, and vice versa. Unlike direct or group conversations, all messages in record conversations are comments or work notes.

Note: By default, record conversations are enabled for all tables that extend Task [task]. Administrators can enable record conversations for additional tables or disable them if desired.

1. Navigate to a task record.
2. Do one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Follow the record</strong></td>
<td>1. In the form header, click <strong>Follow</strong>.</td>
</tr>
<tr>
<td></td>
<td>The system adds you as a member of the record conversation, but does not open the conversation. You can open the conversation from the Connect sidebar.</td>
</tr>
<tr>
<td><strong>Follow the record and open a chat mini window</strong></td>
<td>1. In the form header, click the down arrow in the <strong>Follow</strong> button.</td>
</tr>
<tr>
<td></td>
<td>2. Select <strong>Open Connect Mini</strong>.</td>
</tr>
<tr>
<td></td>
<td>The system adds you as a member of the record conversation and opens it in a Connect mini window.</td>
</tr>
<tr>
<td><strong>Follow the record and open the Connect workspace</strong></td>
<td>1. In the form header, click the down arrow in the <strong>Follow</strong> button.</td>
</tr>
<tr>
<td></td>
<td>2. Select <strong>Open Connect Full</strong>.</td>
</tr>
<tr>
<td></td>
<td>The system adds you as a member of the record conversation and opens it in the Connect workspace, the full-screen view.</td>
</tr>
</tbody>
</table>

The **Follow** button is relabeled **Following**.

3. Add comments or work notes to the record directly from Connect.

Any comments or work notes added to the record appear in the record conversation. Likewise, comments and work notes you add to the conversation appear on the record.

a) In the record conversation, by the text entry field, click the message type icon (_comments). |

b) Select **Comment** or **Work Note**.

c) Enter a message.

By default, record conversation messages are added as comments.

**Note:** If you add an attachment to a record conversation, it is not attached to the record. The file name appears in the comment or work note on the record.

*Edit your notification settings* for the record conversation.

**Edit basic conversation details**

In Connect Chat, you can customize the avatar, name, and description for a group or record conversation.

Role required: none

These details are not editable for direct conversations.
Note: The name and description for a record conversation default to the record name and short description. However, editing the name or description for a record conversation does not change anything on the underlying record.

1. Open the a group or record conversation in the Connect workspace.
2. Click anywhere in the conversation header to the left of the icons.
   A pop-up appears to edit conversation details.
3. Edit the conversation image, name, and description as needed.
4. Click Save.

Add or remove conversation members

In Connect Chat, any member of a group or record conversation can add or remove other conversation members.

Role required: none

1. Open a group or record conversation in the Connect workspace.
2. In the conversation tools to the right of the conversation pane, click the member list tab ( ).
3. Do one of the following actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a member</td>
<td>1. Click Add Member to Group.</td>
</tr>
<tr>
<td></td>
<td>2. Use the search field to find and select a user.</td>
</tr>
<tr>
<td>Remove a member</td>
<td>1. Point to a member name.</td>
</tr>
<tr>
<td></td>
<td>2. Click the minus icon ( - ).</td>
</tr>
</tbody>
</table>

Create a Connect conversation from a Visual Task Board

Create a Connect Chat conversation from a Visual Task Board to collaborate with board members and keep track of activity as it happens.

Role required: none

When you create a conversation from a task board, all the board members become members of the conversation. Adding or removing users from the board members list automatically adds or removes them from the conversation. However, adding or removing users from the conversation does not add or remove them from the board members list.

2. Open a task board.
3. In the board header, click the connect with board members icon ( ).
   The system automatically opens a record conversation for the board. Each board member becomes a member of the conversation.

Connect Chat administration

Administrators can configure various performance settings and features of Connect Chat.
Configure the email notification interval

The email notification interval determines how long the system waits before sending a Connect Chat email notification to an inactive user.

Role required: admin

The default interval is 3 minutes. You can change this value.

1. Navigate to `sys_properties.list`.
2. Locate the `collaboration.email_interval` property.
3. Set the Value to a different number of minutes.

Enable record conversations for a table

Record conversations are enabled for all tables that extend Task [task] by default. You can configure record conversations for additional tables.

Role required: admin

To create record conversations, users must have access to the Follow button in the form header. To show the button and thus enable record conversations for a table, the following conditions must be true.

- The `live_feed` dictionary attribute must be set to `true` for the table.
  - By default, `live_feed=true` for all tables that extend Task [task].
- The `glide.live_feed.task_header_button` property must be set to `both` or `collaboration`.
  - By default, `glide.live_feed.task_header_button` is set to `collaboration`.

1. Navigate to the list view of a table for which you want to enable record conversations.
2. Right-click any column header.
   The list of dictionary entries for the table appears.
4. Open the dictionary entry that has Type set to Collection.
   The Dictionary Entry form appears.
5. In the Attributes related list, click New.
   The Dictionary Attribute form appears.
6. In the Attribute field, enter Live feed.
7. In the Value field, enter true.
8. Click Submit.
9. Navigate to `sys_properties.list`.
10. Locate the `glide.live_feed.task_header_button` property.
11. Ensure the Value is set to both or collaboration.

The `glide.live_feed.task_header_button` property also controls whether the show live feed icon (_Show Live Feed_) appears. If the property Value is collaboration, the show live feed icon does not appear on form headers.
Prevent users from following records in Connect for all tables

Users can follow records in Connect for any table that extends Task [task] by default. You can disable this functionality for all tables.

Role required: admin

To disable record conversations for all tables, prevent the Follow button from appearing in all form headers by editing the glide.live_feed.task_header_button property.

1. Navigate to sys_properties.list.
2. Locate the glide.live_feed.task_header_button property.
3. Set the Value to live_feed.

Prevent users from following records in Connect for a specific table

Users can follow records in Connect for any table that extends Task [task] by default. You can disable this functionality for a specific table.

Role required: admin

To disable record conversations for a specific table, prevent the Follow button from appearing in form headers for the table. To do so, set the live_feed dictionary attribute to false for the table.

1. Navigate to the list view of a table for which you want to disable record conversations.
2. Right-click any column header.
   The list of dictionary entries for the table appears.
4. Open the dictionary entry that has Type set to Collection.
   The Dictionary Entry form appears.
5. In the Attributes related list, locate the Live feed dictionary attribute.
6. Set the Value to false.

Note that setting the dictionary attribute to false also removes the show live feed icon ( ) for the table.

Connect Support

Connect Support is a real-time messaging tool that enables support agents to easily keep track of their support cases, quickly find solutions, and resolve problems quickly.

Connect Support builds on the messaging platform provided with Connect. For general information about the Connect interface, setup, and administration, see Connect on page 2261. When Connect Support is enabled, users designated as support agents have access to the support tab of the Connect sidebar.

Features include:

- Administrators can create chat queues and enable users to access live support.
- Support agents can monitor the queues to provide instant support.
- Drag-and-drop sharing of links, files, and records.

UI16 or UI15 is required to use Connect Support.

Note:
Monitor incoming Connect Support conversations

In the support tab of the Connect sidebar, you can monitor the queues for which you are an agent and accept incoming conversations.

Role required: none

**Note:** The support tab is visible only if you are an agent for one or more queues.

1. Navigate to **Collaborate > Connect Support**. The Connect workspace opens in a new tab.

2. Click the support tab of the Connect sidebar, indicated by a headset icon (📢). The support tab displays **Queues** to which you belong. It also displays your open support conversations under **Cases**. When a user starts a support conversation or an agent transfers a conversation to a queue, any agent who belongs to the associated queue has the option to accept the conversation. An agent can also request to transfer a conversation directly to you.
3. Accept a conversation in one of the following ways.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept a conversation from a queue</td>
<td>Under Queues, click Accept by the queue. The conversation opens in the</td>
</tr>
<tr>
<td></td>
<td>conversation pane and an entry appears in the Cases section of the sidebar.</td>
</tr>
<tr>
<td>Accept a transfer request</td>
<td>Under Cases, click Accept by a transfer request.</td>
</tr>
</tbody>
</table>

4. Respond to the user and help resolve the issue.

By default, your messages are added to the conversation record as comments and are visible to the user.

Share knowledge in a Connect Support conversation

The support view of the Connect workspace has a built-in knowledge tool that makes it easy to search for knowledge articles and share them in a conversation.

Role required: none

Use the knowledge tool to research user questions and provide solutions quickly and easily.

1. Open a support conversation in the Connect workspace.
   The knowledge tool is not available in Connect mini windows.

2. In the conversation tools area to the right of the conversation pane, click the knowledge base tab, which is represented by an open book icon.

The knowledge homepage appears in the conversation tools pane.
3. Search for articles related to the user's issue.
   For more information about finding knowledge, see Knowledge search on page 2131.

4. To share an article in the conversation, drag a knowledge article link to the conversation pane and drop it anywhere in the drop zone that appears.

Transfer a Connect Support conversation to a different agent or queue

You can transfer a Connect Support conversation to a different agent in the queue or to a different queue.

Role required: none

1. Open a Connect Support conversation.

2. At the bottom of the conversation, click the menu icon (Ξ) to open the Connect actions menu.
3. In the Connect actions menu, select **Transfer**. A transfer dialog box opens.

4. Do one of the following actions.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer the conversation to an agent</td>
<td>1. To start the transfer request, click Transfer by an agent's name. Be sure to choose an agent who is online, which is indicated by a green dot on their avatar. You can cancel the transfer any time before it is accepted.</td>
</tr>
<tr>
<td></td>
<td>2. When the transfer is accepted, select one of the following options in the dialog box that appears. • Stay: Remain in the conversation. This is also known as a warm transfer. This option is ideal if you want to provide the new agent with background information or introduce the user to the new agent. • Leave: Exit the conversation. This is also known as a cold transfer.</td>
</tr>
<tr>
<td>Transfer the conversation to a queue</td>
<td>Click Transfer by a queue. The system automatically removes you from the conversation and the conversation enters the queue you selected.</td>
</tr>
</tbody>
</table>

**Escalate a Connect Support conversation**

If an escalation path is defined for a Connect Support conversation, you can use a shortcut to escalate a Connect Support conversation to a different queue.

Define an escalation path for the queue, using the Escalate to field on the Chat Queue [chat_queue] table. For more information, see Administer Connect Support queues on page 2297.

Role required: none

1. Open a Connect Support conversation.
2. At the bottom of the conversation, click the menu icon (≡) to open the Connect actions menu.

3. In the Connect actions menu, select Escalate.
   A confirmation dialog box appears.

4. In the dialog box, click Escalate.
   The system automatically removes you from the conversation and the conversation enters the escalated queue.

Create an incident from a Connect Support conversation

You can use a shortcut to create an incident on behalf of a user, directly from a Connect Support conversation.

Role required: none

You might want to create an incident if you cannot resolve the user's issue over chat or if you want to create a record of the conversation to share with the user. When you create an incident from a support conversation, the system copies the conversation history to the incident activity stream as comments and work notes. Future messages are tracked in the incident as well.

---

**Note:** Administrators can customize the behavior of the Create Incident Connect action. For more information, see Administer Connect actions on page 2283.

---

1. Open a Connect Support conversation.

2. At the bottom of the conversation, click the menu icon (≡) to open the Connect actions menu.

3. In the Connect actions menu, select Create Incident.
   In the conversation tools area to the right of the conversation, a new incident form opens in a record tab. The system automatically sets the Caller field to the user who opened the support conversation.

4. Complete the form as necessary and click Submit.
   Any comments or work notes in the record conversation appear as comments on the incident form.
   Work notes do not appear in the chat for the ESS user. The chat agent can select whether a message is a Comment or Work Note in the conversation.
   a) In the record conversation, by the text entry field, click the message type icon.

   ![Message Type Icon](image)

   b) Select Comment or Work Note.
   c) Enter a message.

   By default, record conversation messages are added as comments.

   **Note:** If you add an attachment to a record conversation, it is attached to the underlying record as well.

The system automatically shares the record in the conversation, copies the conversation to the record activity stream, and references the record on the Chat Queue Entry [chat_queue_entry] table. Any new journal fields added to the record do not appear in the chat.

Connect Support administration

Administrators can configure various performance settings and features of Connect Support.
Note: There are also settings that apply to both Connect Chat and Connect Support. For more information, see Connect administration on page 2282.

Legacy chat to Connect Support transition

Use these basic steps to transition from legacy chat to Connect Support.

Context

Connect Support is an extension of Connect. Users designated as support agents have access to the support tab of the connect sidebar. Connect Support should not be used concurrently with legacy chat.

Before you activate Connect Support

Inform your support agents to close out any existing chats.

Legacy chat uses a different field to determine which state a help desk chat is in. A fix script is built into the Connect Support plugin in Geneva patch 8 that circumvents this issue by closing out any existing help desk chat sessions. Manually closing out any existing chats before allowing this script to run helps prevent any loss of chat information.

If you do not close out existing help desk chats your metrics and chat queue won't work accurately.

Activate the Connect Support plugin

When you activate the Connect Support plugin, the glide.connect.support.enabled property redirects the legacy chat URL to Connect Support.

For more information, see Activate Connect Support on page 2265.

Hide the Social IT application

Activating Connect Support doesn’t completely eliminate the legacy chat module. To prevent users from using it, hide the Social IT application menu. For more information, see Enable or disable an application menu or module on page 44.

Administer Connect Support queues

You can define the support agents, default messages, schedule, and escalation path for each Connect Support queue.

Role required: admin

Note: Both Connect Support queues and legacy chat queues are stored on the Chat Queue [chat_queue] table. Connect and legacy chat should not be used concurrently.

1. Navigate to Collaborate > Administration > Queues.
2. Click New or open an existing queue from the list.
3. Complete the Chat Queue form, as appropriate.
### Table 556: Chat Queue form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the queue.</td>
</tr>
<tr>
<td>Active</td>
<td>Not used in Connect.</td>
</tr>
<tr>
<td>Assignment group</td>
<td>User group that contains the support staff for the queue. Any user in the group can view the queue in the Connect sidebar and accept chats. No other users can access the queue. This field must be populated.</td>
</tr>
<tr>
<td>Average wait time</td>
<td>Average time it takes for an agent to accept a chat in the queue. This value is automatically calculated and should not be edited manually.</td>
</tr>
<tr>
<td>Confirm problem</td>
<td>Not used in Connect.</td>
</tr>
<tr>
<td>Escalate to</td>
<td>Different queue to which an agent can escalate a chat. For example, there might be a queue for high priority support chats. When a queue is defined in this field, agents can access the Escalate option in conversations.</td>
</tr>
<tr>
<td>Initial agent response</td>
<td>Message that users see when an agent accepts their chat. For example, Thank you for contacting support. We are looking into your question now and will be with you shortly.</td>
</tr>
<tr>
<td>Not available</td>
<td>Message that users see when they attempt to start a chat outside the queue’s defined Schedule. You can use HTML to format the message and include links or media.</td>
</tr>
<tr>
<td>Question</td>
<td>Initial phrase that users see when they start a new chat in the queue. For example, How can I help you?</td>
</tr>
<tr>
<td>Schedule</td>
<td>Schedule that defines when the queue is available. Leave the field blank to make the queue available all the time. Users cannot start a new conversation in the queue outside the schedule hours.</td>
</tr>
</tbody>
</table>

4. To review conversations associated with the queue, scroll to the Chat Queue Entries related list and review the records for the following information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long the user waited and the result</td>
<td>Review information in the Action and Wait time columns. Abandoned in the Action column indicates the chat was completed.</td>
</tr>
<tr>
<td>Who is actively helping employees in the queue</td>
<td>Review the names of HR agents in the Assigned to column.</td>
</tr>
</tbody>
</table>
Make a Connect Support queue accessible to end users

To make a Connect Support queue accessible to end users, use the accepted URL format.
Create a queue. Create agents for the queue by assigning users to the assignment group associated with the queue.
Role required: admin
For example you might create a module or add a link to a portal. The accepted URL format is https://<instancename>.service-now.com/$chat_support.do?queueID=<sys_id>.

1. Navigate to Collaborate > Administration > Queues.
2. Right-click the name of the queue to which you want to link.
3. In the context menu, select Copy sys_id.
4. Copy the sys_id.
5. Preview the support queue by navigating to https://<instancename>.service-now.com/$chat_support.do?queueID=<sys_id>.
6. Create a module or other link to the queue using the URL.

Configure Connect Support chat timeout

Use system properties to determine when support sessions time out after being idle.
Role required: admin
1. Navigate to the sys_properties table.
2. Search for the following properties and edit the number in the Values field.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect.support.idle.count_down</td>
<td>Number of seconds to count down from before marking end-user as having left their support session.</td>
</tr>
<tr>
<td>connect.support.idle.delay</td>
<td>Number of seconds to wait without user interaction, before presenting end-users with an idle countdown timer.</td>
</tr>
</tbody>
</table>

Legacy chat

Legacy chat provides real-time communication via instant messaging between users in a ServiceNow instance.

**Note:** The Connect on page 2261 feature provides an updated instant messaging platform, which is activated by default for new instances. The legacy chat and Connect features should not be used concurrently. There is no migration path from legacy chat to Connect.

Features include:

- One-to-one chats (instant messaging) between users.
- Chat rooms for conversations with multiple users. Chat rooms may be public (any user can join) or private (only invited users can join).
Chat rooms linked to task records. Users can work together to solve issues, and conversation history can be shared by everyone who needs to reference it.

Help desk chat. End users can access live support via instant messaging. Service desk staff can resolve basic issues in real-time or create incidents directly from chat requests for more extensive issues.

Figure 610: Chat Desktop

Get started with the legacy chat feature

Get started with legacy chat.

Role required: admin

1. Establish use guidelines. Social media can improve communication and aid productivity. To get the most out of these tools, establish clear and simple social media guidelines that foster information sharing and a comfortable work environment.
2. Activate the Chat plugin to enable the legacy chat and legacy Help Desk Chat features.

3. Configure security settings. Users must log in to use the legacy chat features in the ServiceNow ITSA Suite. The standard system security settings and Social IT-specific security settings are available. To learn more about setting up these features, refer to the plugin activation pages.
   - Chat rooms may be Public (any user can join) or Private (only invited users can join).
   - Administrators can limit who can read chat messages on tasks and who can create chat rooms.

Activate legacy chat

You can activate the Chat plugin within the instance if you have the admin role.

Role required: admin

Before activating the Chat plugin, consider the installed components, dependencies, and impact.

- Installed Components: Include tables, a field, business rules, a script include, an application, a user role, properties, an event, and an email notification. For more information, review the components that are installed with chat.
- Dependencies (installed automatically): Social IT Infrastructure.
- Impact: The plugin installs new features; it does not overwrite or impact current configurations. It has minimal impact on the system. However, when the system is configured to use short polling (see the Properties in Installed with legacy chat on page 2301) and the client is in debug mode, users may experience a performance impact. Polling also keeps the session alive when the chat desktop is open.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Installed with legacy chat

What components are installed with legacy chat.

Demo data is available for legacy chat. The demo data creates a chat queue called Help Desk Chat that is supported by the assignment group Chat Support. Additionally, the Help Desk Chat link is added to the header of the Employee Self-Service portal. To learn more, see Set Up Chat Queues for Help Desk Chat.

Other:
- Event: A new event called chat.invite is registered to trigger an email notification when users are invited into a chat room.
- Email Notification: An email notification called Chat Room Invite is added to send when users are invited to a chat room.
Fields installed with legacy chat

Legacy chat adds the following fields.

<table>
<thead>
<tr>
<th>Display name [Table name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header [content_block_header]</td>
<td>Adds the Help Desk Chat link to the ESS portal page.</td>
</tr>
</tbody>
</table>

Properties installed with legacy chat

Legacy chat adds the following properties. Making changes to legacy chat properties does not affect Connect Chat.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.chat.invite_fields | Comma-separated list of fields used to generate the invites when creating a chat room from a task. The user is presented with check boxes for each of the specified fields to select the invites for the chat room. The fields should be references or glide_lists of sys_user or sys_user_group.  
  - **Type**: string  
  - **Default value**: assignment_group,watch_list  
  - **Location**: Social IT > Chat Administration > Properties |
| glide.chat.show_emoticons | Setting that determines whether to display emoticons in conversations.  
  - **Type**: true | false  
  - **Default value**: true  
  - **Location**: Social IT > Chat Administration > Properties |
| glide.chat.sound.message_received.mp3 | Sound played when a message is received in chat (mp3).  
  - **Type**: string  
  - **Default value**: media/rcvmsg.mp3x  
  - **Location**: Social IT > Chat Administration > Properties |
| glide.chat.sound.message_received.ogg | Sound played when a message is received in chat (ogg).  
  - **Type**: string  
  - **Default value**: media/rcvmsg.oggx  
  - **Location**: Social IT > Chat Administration > Properties |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.chat.sound.queue_beep.mp3 | Sound played when a new user enters the chat queue (mp3). Both this property and glide.chat.sound.queue_beep.ogg must be defined for either property to work.  
  • **Type**: string  
  • **Default value**: media/button_toggle_on.mp3x  
  • **Location**: Social IT > Chat Administration > Properties |
| glide.chat.sound.queue_beep.ogg | Sound played when a new user enters the chat queue (ogg).  
  • **Type**: string  
  • **Default value**: media/button_toggle_on.oggx  
  • **Location**: Social IT > Chat Administration > Properties |
| glide.chat_room.create_roles | Comma-separated list of roles that are allowed to create chat rooms.  
  • **Type**: string  
  • **Default value**: itil  
  • **Location**: Social IT > Chat Administration > Properties |
| glide.short_poll_delay | Short polling delay for XMPP requests. Enter a value in milliseconds. The minimum value is 250. With short polling, the browser sends a request to the server in fixed intervals defined by the property. To minimize performance impact, it is recommended that this value is set greater than or equal to the default value.  
  • **Type**: integer  
  • **Default value**: 1000 (one second)  
  • **Location**: Social IT > Chat Administration > Properties |

### Script includes installed with legacy chat

Legacy chat adds the following script includes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChatUtils</td>
<td>Provides utilities for chat conversations.</td>
</tr>
</tbody>
</table>

### Tables installed with legacy chat

Legacy chat adds the following tables.
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chat Actions</td>
<td>Defines additional chat window menu items. Access at Social IT &gt; Chat Administration &gt; Actions (administrator and chat_admin).</td>
</tr>
<tr>
<td>[chat_actions]</td>
<td></td>
</tr>
<tr>
<td>Chat Channel</td>
<td>Stores chat channels, which are chats between members. This table is extended by Chat thread [chat_thread] (a one-to-one chat) and chat_room (a multi-user chat). Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_channel]</td>
<td></td>
</tr>
<tr>
<td>Chat Channel Member</td>
<td>Relationship table that associates channels and members. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_channel_member]</td>
<td></td>
</tr>
<tr>
<td>Chat Message</td>
<td>Stores instant messages sent from any chat window. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_message]</td>
<td></td>
</tr>
<tr>
<td>Chat Presence</td>
<td>Stores presence (status) information for a user. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_presence]</td>
<td></td>
</tr>
<tr>
<td>Chat Queue</td>
<td>Stores the groups and schedules for providing support via instant message. Defined at Social IT &gt; Chat Administration &gt; Queues (administrator and chat_admin).</td>
</tr>
<tr>
<td>[chat_queue]</td>
<td></td>
</tr>
<tr>
<td>Chat Queue Entry</td>
<td>Stores user requests for live support. Extends Task [task]. On a chat queue record, the Chat queue entries related list displays the entries that are associated with that chat queue. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_queue_entry]</td>
<td></td>
</tr>
<tr>
<td>Chat Room</td>
<td>Stores chat rooms (multi-user chat). Extends Chat channel [chat_channel]. Can be modified if necessary (such as to change a chat room from public to private), but should almost always be managed by system functionality.</td>
</tr>
<tr>
<td>[chat_room]</td>
<td></td>
</tr>
<tr>
<td>Chat Roster</td>
<td>Maintains users that are associated with a user’s My Friends list. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_roster]</td>
<td></td>
</tr>
<tr>
<td>Chat Roster Member</td>
<td>Tracks users that are members of a chat room. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_roster_member]</td>
<td></td>
</tr>
<tr>
<td>Chat Thread</td>
<td>Stores one-to-one chats. Extends Chat channel [chat_channel]. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[chat_thread]</td>
<td></td>
</tr>
<tr>
<td>Live Headline</td>
<td>Sends system messages. For example, the system may send a message when a user enters or leaves a chat room. Extending or modifying data in this table is not recommended.</td>
</tr>
<tr>
<td>[live_headline]</td>
<td></td>
</tr>
</tbody>
</table>
## Live Poll Message
**[sys_live_message]**
System table that stores and manages chat messages. Extending or modifying data in this table is not recommended.

## Live User
**[sys_live_user]**
System table that stores and maintains chat users. Extending or modifying data in this table is not recommended.

## Stores the last live sequence number
**[sys_live_last_sequence]**
Stores the last sequence ID. Extending or modifying data in this table is not recommended.

### Business rules installed with legacy chat

Legacy chat adds the following business rules.

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chat Queue [Task.active] Updater</td>
<td>Chat Queue Entry [chat_queue_entry]</td>
<td>Sets the chat queue entry Action field to Waiting when the action changes.</td>
</tr>
<tr>
<td>SNC - Chat Queue Average Wait Time</td>
<td>Chat Queue Entry [chat_queue_entry]</td>
<td>When a chat queue entry is accepted, calculates the average wait time for the queue by averaging the last 20 chat queue entries.</td>
</tr>
</tbody>
</table>

### Legacy chat administration

Users with the chat_admin role can administer various aspects of legacy chat.

#### Define chat room access rights

Role required: chat_admin

To define the access rights for creating chat rooms:

1. Navigate to **Social IT > Chat Administration > Properties**.
2. Locate the property **Comma-separated list of roles that are allowed to create chat rooms**.
3. Enter user roles. A user must have one of the specified roles to create a chat room.

#### Change chat room message read access

In legacy chat, you can associate a chat room with a record, such as an incident, and add the chat messages to a **Chat Activity** journal field on the form.

Role required: admin

By default, a user can read the chat messages for a room if either of the following is true:

- The room is public
• The user is a member of the room

To change chat room message read access:
  Modify the Chat message access control rule or create a new rule.
  Read access to the messages displayed is handled by the access control list security operation chat_messages_read on the Chat room [chat_room] table.

View a legacy chat message as a journal field

Chat rooms can be linked to any task record in the following ways:
• Create a room from the record.
• Create a record from a chat room, such as creating an incident from a Help Desk Chat.

To view the chat messages as a journal field:
  Configure the task record form to select the Chat Activity field.
  The maximum number of chat messages that are displayed by this field is 1000.

![Chat Activity](image)

Delete a chat room

You can delete chat rooms in legacy chat.
Role required: chat_admin
Note that when you delete a chat room, the system also deletes the records for any chat members and messages.

1. In the application navigator filter, enter chat_room.list.
2. Delete multiple chat rooms from the list or open a chat room record and click Delete.

Chat actions

Chat actions are additional items that appear in the chat window menu (requires the Chat plugin). Administrators and users with the chat_admin role can define chat actions.

Add actions to the legacy chat window menu

1. Navigate to Social IT > Chat Administration > Actions.
2. Select an action to edit or click New.
3. Enter the chat action details and click Update or Submit.

The following example demonstrates how to add a chat action that appears only on windows where the user is a Help Desk Agent. The action appears on the menu as Show User Type and opens a popup window that indicates a current user of queue_agent.

- **Action Name**: Show User Type
- **Active**: Select the check box.
- **Order**: 100
• **Client Show Condition:**

```javascript
answer = g_chat.getChatType() == 'queue_agent';
```

• **OnClick Action Script:**

```javascript
alert('The current user is: ' + g_chat.getChatType());
```
Chat action details

Chat action details

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Name</td>
<td>Enter a label for the action as it appears in the menu.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate or clear the check box to deactivate the action. Only active actions may appear in the menu.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter the order in which the action appears in the menu. Standard menu items (such as Show Timestamps) always appear at the top of the chat menu, followed by chat actions in order.</td>
</tr>
<tr>
<td>Client Show Condition</td>
<td>Define the conditions under which this menu option appears. The show condition must set the pre-defined answer variable to a boolean value.</td>
</tr>
</tbody>
</table>
Available methods associated with the g_chat variable

The g_chat variable is a legacy chat window object that is available in the **Client Show Condition** and **OnClick Action Script** of a chat action definition.

The following table defines the available methods associated with this object.

**Method Summary**

<table>
<thead>
<tr>
<th>Return Object</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td><code>getActiveUsers()</code>&lt;br&gt;Returns the active users.&lt;br&gt;Returns:&lt;br&gt;Array - Returns the active users.</td>
</tr>
<tr>
<td>String</td>
<td><code>getChannelJID()</code>&lt;br&gt;Returns the channel JID, which is a sys_id for a record chat_channel table.&lt;br&gt;Returns:&lt;br&gt;String - Returns the channel JID.</td>
</tr>
<tr>
<td>String</td>
<td><code>getChatType()</code>&lt;br&gt;Determines the chat window type.&lt;br&gt;Returns:&lt;br&gt;String - Returns one of the following values that specifies the type of user:&lt;br&gt;• queue_agent: Help Desk Chat support agent&lt;br&gt;• queue_user: Help Desk Chat end user&lt;br&gt;• group_chat: The individual is a member of a chat room (multiple users)&lt;br&gt;• conversation: The individual is a member in a private chat with another user</td>
</tr>
<tr>
<td>String</td>
<td><code>getChatQueueAgent()</code>&lt;br&gt;Returns the sys_id for agent that is administering this thread.&lt;br&gt;Returns:&lt;br&gt;String - Returns the sys_id of the chat queue agent.</td>
</tr>
<tr>
<td>String</td>
<td><code>getChatQueueUser()</code>&lt;br&gt;Returns the sys_id for end user of the chat queue.</td>
</tr>
</tbody>
</table>
### Legacy chat use

Legacy chat allows you to interact and collaborate through real-time instant messages with other users in an instance.

Use legacy chat to:

- Initiate or participate in chat conversations with one or multiple users.
- Initiate or participate in chat conversations that are attached to task records (such as an incident).
- See when other users are available to chat.

To open the chat desktop, navigate to **Social IT > Chat**.

---

**Note:** Service desk staff may also provide live support to other users via help desk chat.
Figure 613: Chat desktop

Update your profile

Your profile identifies your contributions to legacy chat conversations.

Your profile identifies your contributions to conversations. It is created automatically the first time you use chat and consists of:

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>From your user account in ServiceNow. Your name appears on your messages and in the member list of any chat room you join.</td>
</tr>
</tbody>
</table>
To update your profile picture:

1. Click your picture in the top of the favorites list.
2. Browse to the desired picture file and click **Open**.

**Note:** Changes to your profile picture affect legacy chat and live feed if the Live Feed plugin is activated.

---

**Figure 614: Chat profile picture**

---

**Update your status**

In legacy chat, your status lets other users know whether you are available to chat.

Your status lets other users know whether you are available to chat. Your current status is indicated by the color of the icon in the upper right of your favorites list, beside your name. View the status of other users in your favorites list.

To change your status:

1. Click the status icon beside your name.
2. Select a status (see table).
Table 558: Change status

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
<th>Appears to others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online</td>
<td>Indicates that you are available to chat.</td>
<td>Green icon:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Beth Anglin</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Figure 616: Chat available</a></td>
</tr>
<tr>
<td>Away</td>
<td>Indicates that you are not available because you are away.</td>
<td>Red icon:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Beth Anglin</a> Away</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Figure 617: Chat away</a></td>
</tr>
<tr>
<td>Away with a message</td>
<td>Indicates that you are not available and gives a reason:</td>
<td>Red icon and message:</td>
</tr>
<tr>
<td></td>
<td>• In a meeting</td>
<td><a href="#">Beth Anglin</a> In a meeting</td>
</tr>
<tr>
<td></td>
<td>• On the phone</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Out to lunch</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You cannot create a custom message from the chat desktop.</td>
<td></td>
</tr>
</tbody>
</table>

[Figure 615: Change status](#)

![System Administrator](image)
<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
<th>Appears to others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invisible</td>
<td>Indicates that you are not available. You can send and receive messages when your status is Invisible. You appear as Offline to other users.</td>
<td>Black icon: Beth Anglin</td>
</tr>
<tr>
<td>Offline</td>
<td>Indicates that you are not available. You cannot send or receive messages when your status is Offline. Users that are not logged in appear as Offline.</td>
<td>Black icon: Beth Anglin</td>
</tr>
</tbody>
</table>

**Use your favorites list**

In legacy chat, your favorites list appears on your chat desktop and provides certain functions.
Table 559: Favorites list

- Users:
  - *Add another user* in the system to your favorites list.
  - See if your favorite users are available to chat. Users are organized alphabetically and in sections by status.
  - Start a conversation by double-clicking a name.
  - Expand or collapse a section by clicking **Online** or **Offline**.

- Rooms:
  - See the list of *your favorite chat rooms*.
  - Join a chat room by double-clicking its name.

- Favorites list toolbar:
  - Options ( ): access a menu with options to *add a favorite user*, *view online users*, *create a chat room*, and *view chat rooms*.
  - Add User ( ): *add a favorite user*.
  - Create Room ( ): *create a chat room*.
  - Invitations ( ): respond to invitations to *join chat rooms*.
To expand or collapse a section in the favorites list, click the section header or click the arrows
(▲)

on the section header.

Add a favorite user

How to add a user to your favorite users list in legacy chat.

1. Click the **Add User** button.

![Add friend](image)

**Figure 626: Add friend**

2. Begin typing a user name and select a user from the list, or click the reference lookup and select a user from the table.

![Search icon](image)

**Figure 627: Search icon**

3. Click **OK**.

Remove a favorite user

How to remove a favorite user in legacy chat.

To remove a user from your favorite users list, right-click the user's name and select **Remove From List**.

View an online user

How to view a list of all users who are available to chat (status of Online) in legacy chat.

1. Right-click the Users section header or click **Options** on the toolbar.

![Chat window menu](image)

**Figure 628: Chat window menu**

2. Select **Show Online Users**.

- Start a one-to-one chat with a user on the list by double-clicking a name.
- **Send Message** or **Add To Friend List** by right-clicking a name.
View a room

How to view a list of all public chat rooms in legacy chat.

1. Right-click the **Rooms** section header or click **Options** on the toolbar.

   ![Chat window menu](image)

   **Figure 630: Chat window menu**

2. Select **Public Rooms**.
   - To join a room, double-click a name or right-click and select **Join Room**. See *Joining Chat Rooms*. 
Start a one-to-one chat

How to start a one-to-one chat in legacy chat.

- To start a chat with one user in your favorites list, double-click the user's name or right-click and select **Send Message**.
- To start a chat with one available user, double-click the user's name on the online users list. See **Viewing Online Users**.

  *Send a message* to start a conversation.

Create a chat room

How to start a chat with multiple users from the legacy chat desktop.

To start a chat with multiple users from the chat desktop (requires access rights):

1. Click **Create Room** on the favorites list toolbar.

   ![Figure 632: Add room](image)

2. Enter the room details (see table).

3. Click **Create Room**. An invitation appears in the favorites list of all invited users and a new chat window opens on your chat desktop.
Create Room

Room name: Chat Room for INC000003

Description: Wireless access not available on floor 3

Password: ********

Room avatar: ![Warning]

Features:
- **Public**: A room that can be found by any user through normal means such as searching and service discovery.
- **Members Only**: A room that a user cannot enter without being on the member list or invited.
- **Temporary**: A room that is destroyed if the last occupant exits.

Invite:
- Fred Luddy
- Beth Anglin
- ITIL User

Select these options:
- Assignment group
- Watch list

Create Room  Cancel

Figure 633: Chat task create
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Room name</td>
<td>Enter a name to appear on the chat window header or click the <strong>Generate a unique room name</strong> button (Figure 634: Room name generator) to use a system-generated name. The default name for a room created from a task is Chat Room for &lt;Task Number&gt;.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a room description that appears under the name in the rooms list. The default description for a room created from a task is &lt;Short Description&gt;.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter a password, if desired. If a password is specified, only users with the password can join the chat room.</td>
</tr>
<tr>
<td>Room avatar</td>
<td>Upload an image that appears beside the room name in the list of rooms. Click the picture and browse to the desired image file.</td>
</tr>
<tr>
<td>Features</td>
<td>Select all applicable check boxes:</td>
</tr>
<tr>
<td></td>
<td>• Public if all users can see the room in the rooms lists.</td>
</tr>
<tr>
<td></td>
<td>• Members Only if only invited users can join the room.</td>
</tr>
<tr>
<td></td>
<td>• Temporary if the room is not saved once all participants have left. This feature is not available for rooms created from tasks; task chat rooms are always available on the task record.</td>
</tr>
<tr>
<td>Invite</td>
<td>Add invited users by using the <strong>glide list</strong>. For a room created from a task, you can also select all applicable check boxes:</td>
</tr>
<tr>
<td></td>
<td>• Assignment group: invite all users in the assignment group for the task.</td>
</tr>
<tr>
<td></td>
<td>• Watch list: invite all users on the watch list for the task.</td>
</tr>
</tbody>
</table>

Create a chat room for a task

How to start a legacy chat with multiple users from a task record.
Note: This procedure is not possible if the Connect on page 2261 feature is enabled.

1. Open the record (example, an incident on which you are working).

![Incident record](image)

2. Click the Create or Join Chat Room related link.

3. If a chat room already exists, click a name to join the existing chat room. If no chat room exists, continue to the next step to create a new room.

![Chat task open](image)

4. Click the create link at the bottom of the window.

5. Enter the room details as you would for a chat room.

6. Click Create Room. An email notification is sent and appears in the favorites list for all invited users (the assignment group and the watch list, if selected, and additional invited users).
7. In the New Room Created message, click **Join Room**.
   - A new chat window opens on your chat desktop.
   - All invited users are listed in the chat member list. Invited users that are not currently participating in the chat are listed in gray.

Join one-to-one chats

In legacy chat, when another user starts a chat by sending a message to you, a chat window opens on your chat desktop. To join the chat, click the window.

Join a chat room

How to join a chat room in legacy chat.

1. If another user invites you to join a chat room, an invitation appears in your favorites list.

   ![Chat invitation](image)

   **Figure 637: Chat invitation**

2. To respond, click the invitation.
   A window opens that displays the room name, room description, and the name of the user who invited you.
   - To join the chat, click **Join**. When the chat window opens on your chat desktop, click the window.
   - To ignore the invitation, click **Decline**.

   ![View invitations](image)

   **Figure 638: View invitations**

To rejoin a favorite chat room that you have left, click the chat room name in the Rooms section of the favorites list.

To join a public chat room without an invitation, right-click the Rooms section header and select **Public Rooms**. Double-click a room name on the list.
Join a chat room for a task

In legacy chat, if another user invites you to join a chat room from a task record, you receive an email notification.

1. Click the link in the email notification.
2. If a prompt for a password appears, enter the password listed in the email notification.
3. When the chat window opens on your chat desktop, click the window.

Join a chat room from a task record without an invitation

How to join a chat room in legacy chat without an invitation.

**Note:** This procedure is not possible if the Connect on page 2261 feature is enabled.

1. Open the task record.
2. Click the Create or Join Chat Room related link.
3. Click a chat room name.
4. If a prompt for a password appears, contact the person who created the chat room for the password.
   If a chat room has a password, only users with the password can join the chat.
5. When the chat window opens on your chat desktop, click the window.

Legacy chat windows

In legacy chat, a chat window appears on your chat desktop for each chat conversation that you start or join.
Send messages in legacy chat

How to send messages in legacy chat.

To chat with other users that have joined the conversation, enter your message in the text field at the bottom of the window and press Enter. The message appears in the other users’ chat windows.

To send a link, type the full URL (example, http://www.service-now.com). When the message is sent, the text appears blue and any user in the conversation can click it to follow the link.

To send a smiley face, type a colon followed by a close parentheses (:)'). When the message is sent, a smiley face 😊 appears in place of the text. For more emoticons, see Sending Emoticons in Messages (Smileys).

Chat activity is indicated with the following alerts:

- When a new message is posted to an inactive browser tab, the tab name blinks.
- When a new message is posted to an inactive chat window, the window header appears in yellow and blinks.
• When a user joins or leaves a room, a message appears in the other room members’ chat windows.

Send emoticons in legacy chat messages (smileys)

How to send emoticons (smileys) in legacy chat.

To send an emoticon (smiley), type text from the following table. When your message is sent, the emoticon appears in place of the text. For example, enter colon followed by a close parentheses (:) and a smiley face 😊 appears in your message.

<table>
<thead>
<tr>
<th>Text</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>:) :-()</td>
<td>😊</td>
</tr>
<tr>
<td>B) B-) BD B-D</td>
<td>😎</td>
</tr>
<tr>
<td>:D :-D</td>
<td>😄</td>
</tr>
<tr>
<td>;* :*-</td>
<td>😞</td>
</tr>
<tr>
<td>:( ::(</td>
<td>😞</td>
</tr>
</tbody>
</table>

Figure 640: Smiley
Figure 641: Shades
Figure 642: Big smile
Figure 643: Kiss
Figure 644: Frown
<table>
<thead>
<tr>
<th>Text</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;3</td>
<td>❤️</td>
</tr>
<tr>
<td>Figure 645: Heart</td>
<td></td>
</tr>
<tr>
<td>;) :-</td>
<td>😄</td>
</tr>
<tr>
<td>Figure 646: Wink</td>
<td></td>
</tr>
<tr>
<td>:P :-P</td>
<td>😐</td>
</tr>
<tr>
<td>Figure 647: Playful</td>
<td></td>
</tr>
<tr>
<td>:O :-O</td>
<td>😳</td>
</tr>
<tr>
<td>Figure 648: Surprise</td>
<td></td>
</tr>
<tr>
<td>X( X-(</td>
<td>😁</td>
</tr>
<tr>
<td>Figure 649: Angry</td>
<td></td>
</tr>
<tr>
<td>:) :&quot;&gt;</td>
<td>😞</td>
</tr>
<tr>
<td>Figure 650: Blush</td>
<td></td>
</tr>
<tr>
<td>:(( :-(</td>
<td>😢</td>
</tr>
<tr>
<td>Figure 651: Crying</td>
<td></td>
</tr>
<tr>
<td>(A)</td>
<td>😎</td>
</tr>
<tr>
<td>Figure 652: Halo</td>
<td></td>
</tr>
<tr>
<td>:? :-?</td>
<td>😇</td>
</tr>
<tr>
<td>Figure 653: Undecided</td>
<td></td>
</tr>
</tbody>
</table>
Change the display

How to change the display of legacy chat.
To view the chat window menu, click the gear button ()

Figure 654: Chat window menu

To show or hide timestamps, select **Show Timestamps** from the chat window menu or press **F2**.
To show or hide the chat member list, select **Show Members** from the chat window menu or press **F4**.

To resize the window, point to the lower corner (left or right). When the pointer changes shape, drag the window to the desired size.
To move the window, point to the header. When the pointer changes shape, drag the window anywhere on your chat desktop.
To close a chat window, click the X in the upper right corner.

Invite a user into a chat

How to invite another user into a chat in legacy chat.

1. Select **Invite User** from the chat window menu.

Figure 655: Chat window menu

2. Begin typing a user name and select a user from the list, or click the reference lookup and select a user from the table.

Figure 656: Search icon

3. Click **OK**.
   An invitation is sent to the selected user. One-to-one chats are automatically converted into temporary chat rooms.

Add a favorite room

How to add a favorite room in legacy chat.

To add a room to your favorites, click the chat window and select **Add To Favorites** from the chat window menu.

Figure 657: Chat window menu
Remove a favorite room

How to remove a favorite room in legacy chat.

To remove a room from your favorites, right-click the room name and select Remove From List.

Set your chat preferences

Use the Chat Window Preferences to set audio notifications and default chat window display options in legacy chat.

Use the Chat Window Preferences to set audio notifications and default chat window display options. The default display options control how a chat window looks when you first open it. You can still control the display options for an individual window as described in Changing the Display.

To view your chat preferences, click your name in the upper left corner of the chat desktop and select Chat Preferences.

![Chat Window Preferences](image)

Figure 658: Chat preferences

To set your preferences, select or clear the check boxes next to the options, then click Update.

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Turn audio notifications on or off.</td>
</tr>
<tr>
<td>Conversation Windows</td>
<td>Set default options for showing timestamps and members in one-to-one chats.</td>
</tr>
<tr>
<td>Group Chat Windows</td>
<td>Set default options for showing timestamps members in chat rooms.</td>
</tr>
</tbody>
</table>
Help desk chat

Communicate with service desk staff using instant messaging.

**Note:** The Connect Support on page 2290 feature provides an updated messaging platform similar to help desk chat. The help desk chat and Connect Support features should not be used concurrently. There is no migration path from help desk chat to Connect Support.

Users can communicate directly with service desk staff using instant messaging in a ServiceNow instance (requires the Chat plugin.)

- Users access live support from the Employee Self-Service portal.
- Service desk staff provide support from the chat desktop.

Use help desk chat to get support

You can contact service desk staff.

2. Log in to your ServiceNow account. Only logged in users can use chat.
3. In the upper right, click the **Service Desk Chat** button. A chat window opens.
4. Enter your question in the text field at the bottom of the window and press **Enter**.
   1. A message confirms that you have entered the chat queue and indicates your position and estimated wait time.
   2. When a service desk staff member accepts your chat and begins working on your question, another message appears.
5. Chat with your service desk agent via instant messaging.

service desk staff may lead you through troubleshooting, ask clarifying questions, or create an incident record to address your question.

**Note:** Administrators can configure Service Desk Chat to be accessible on content management (CMS) pages.
Use help desk chat to provide support

Staff who are assigned to chat support can provide live support via instant messaging. Administrators and users with the chat_admin role can assign chat support staff (see Set Up Chat Queues for Help Desk Chat).

To provide users with live support using chat:

1. Navigate to Social IT > Chat to open your chat desktop.
   - A chat queue agent window appears on your chat desktop for each chat queue you are assigned to. If no agent windows appear, ask an administrator to add you as a member of the assignment group for a chat queue.
   - The chat queue agent window displays the number of users in the queue and how long they have been waiting.
   - Every member of the assignment group sees the associated chat queue agent window.

2. Click Answer Next User to answer the next user or click Answer beside a specific user in the queue. A chat window opens.
   - Only the agent and the user can see the chat window.

Provide support via instant messaging. Chat queue agents can access additional support functions in the chat window menu:

- **Invite User**: invite another expert to assist with a support chat. An invitation appears in the favorites list of an invited user.
- **Create Incident from Chat**: create a new incident record with the initial question as the short description and the support chat as a linked chat room (accessible from the Create or Join Chat Room related link on the Incident form).
Geneva  ServiceNow  ServiceNow Platform

Note: Administrators can edit this action or add additional actions to the chat window menu.

Figure 660: Chat Queue Agent

Set up chat queues for help desk chat

Help desk chat allows users to communicate directly with Service Desk staff via instant messaging in an instance.

Chat queues define the schedules, staff, and system messages for help desk chat. Administrators and users with the chat_admin role can set up chat queues.

Define a chat queue for legacy chat

How to define a chat queue for legacy chat.

Note:

The Chat Queue [chat_queue] table is also used by Connect Support on page 2290. Legacy chat and Connect Support should not be used concurrently.

1. Navigate to Social IT > Chat Administration > Queues.
2. Select a chat queue to edit or click New.
3. Enter the chat queue details and click Update or Submit.
### Table 563: Chat queue details

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name that end users see as a title for the queue.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to activate or clear the check box to deactivate the chat queue.</td>
</tr>
<tr>
<td>Assignment group</td>
<td>Select the group that contains the support staff for the chat queue. Every agent sees the chat queue on their chat desktop and can answer any user that is waiting in the queue. To assign staff members to the group, see <a href="#">Assign Service Desk Staff to a Chat Queue</a>.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Select a schedule that defines when a queue is available (see <a href="#">Use Schedules</a>). If the queue is always available, clear the field.</td>
</tr>
<tr>
<td>Question</td>
<td>Enter the initial question that end users see when they open a new Help Desk Chat. For example, How can I help you?</td>
</tr>
<tr>
<td>Confirm problem</td>
<td>(Optional) Enter the confirmation message that end users see when they enter an initial question. For example, Thank you for contacting support. Your problem has been submitted and an agent will be with you shortly.</td>
</tr>
<tr>
<td>Initial agent response</td>
<td>(Optional) Enter the message that end users see when an agent accepts their chat. For example, Thank you for waiting. I am looking into your question now and will be with you shortly.</td>
</tr>
<tr>
<td>Not available</td>
<td>Enter the message that end users see when a Help Desk Chat request is not accepted because the queue is outside its scheduled availability (defined by the Schedule field).</td>
</tr>
</tbody>
</table>

---

**Assign service desk staff to a chat queue**

Chat queue agents are Service Desk staff that are members of a chat queue's assignment group.

To assign agents to a chat queue:

1. Navigate to **User Administration > Groups**.
2. Select an existing assignment group or click **New**.
3. Enter or modify the group details (see [Create a Group](#)).
4. Right-click the header and select **Save**.
5. In the Group members related list, click **Edit**.
6. Using the **slushbucket**, add support staff to the group.
7. Navigate to **Social IT > Chat Administration > Queues**.
8. Select the chat queue for which the group provides support.
9. In the Assignment Group field, select the group.

![Chat queue group](image)

Make help desk chat accessible to an end user

Users access live support from the Employee Self-Service Portal (ESS Portal). To make a Help Desk Chat queue accessible to end users, add a link to the ESS Portal (requires the **Content Management Plugin**).

To change the default link in the ESS header:

1. Navigate to **Content Management > Headers**.
2. Open **Portal - Header**.
3. In the Chat queue field, enter the chat queue that users access by clicking the link in the ESS header. Clear the field to remove the chat queue link.
4. Click **Update**.
Add a custom link elsewhere on the ESS portal
How to add a custom help desk chat link on the ESS portal.

1. Navigate to Content Management > Static HTML.
2. Click New.
3. Enter a Name.
4. In the Static content field, click Toggle HTML Source to edit the field in HTML source mode.

Figure 663: HTML editor button

5. Create a link to the desired Help Desk Chat queue (see Link Syntax).
6. Click Submit.
7. Add the new content block to the desired location on the portal (see Adding Content to the Page).

Link syntax
To open a Help Desk Chat queue from a custom link on the ESS Portal, you must call the CustomEvent.fire method using appropriate arguments.

To open a Help Desk Chat queue from a custom link on the ESS Portal, you must call the CustomEvent.fire method using appropriate arguments. Details of the API call and an example that generates an anchor link are provided below.

API Call:

```javascript
CustomEvent.fire(LiveEvents.LIVE_EVENT,
    LiveEvents.LIVE_WINDOW_JOIN_QUEUE_QUERY, CHAT_QUEUE_SYS_ID,
    CHAT_QUEUE_SYS_NAME);
```

where the variables are:

- CHAT_QUEUE_SYS_ID: Specifies the sys_id for the chat queue.
• **CHAT_QUEUE_SYS_NAME**: Specifies the name of the chat queue (must match the value the Name field of the chat queue record).

Example: This example creates an anchor link that opens the Help Desk Chat queue.

```html
<a href="#" onclick="CustomEvent.fire(LiveEvents.LIVE_EVENT,
    LiveEvents.LIVE_WINDOW_JOIN_QUEUE_QUERY,
    'c54f0abf0a0b452db84664f409c79c',
    'Help Desk Chat'); return false;">
    Help Desk Chat
</a>
```

**Monitor chat queues**

Chat queues can yield useful Key Performance Indicators (KPI) for evaluating support effectiveness.

- **Queue Wait Time**: amount of time a user waits in the queue before a help desk agent accepts the request.
- **Percentage of Chats Abandoned**: users that exit the queue before an agent responds (user stopped waiting).
- **Percentage of Chats Accepted**: requests that are answered by an agent.

---

**Note**: This information is not calculated automatically. Administrators may calculate these values based on data collected by chat queues.

**Monitor help desk chat tasks**

Help Desk Chat requests are tracked in the Chat queue entries table, which appears as a related list on the associated chat queue record.

Help Desk Chat requests are tracked in the Chat queue entries table, which appears as a related list on the associated chat queue record. Because this table extends the Task table, administrators can take advantage of core task functionality. For example, administrators can attach Service Level Agreements (SLAs) to chat queue entries to evaluate the response time of Help Desk Chat support.

---

**Live Feed**

Live Feed is a social IT application that provides a place to post and share content in a ServiceNow instance.

Live Feed content forms a searchable knowledge source for sharing information within an organization. Main features include the following:

- Users can post, reply to, and rate messages, including links and images.
- Administrators can set up automatic messages that are generated when specific records are updated throughout the system.
- Users can subscribe to receive email notifications when new messages are posted.

Live feed includes different types of feeds.
Note: Live Feed v2 is active for all new instances by default. If you are upgrading from an earlier version of ServiceNow, you need to activate live feed v2 to use these features. Refer to Legacy: Live Feed for more information about Live Feed v1.

Administer Live Feed

Administrators and users with the live_feed_admin role can configure live feed security and provide access to live feed from content management pages. Administrators can also set up live feed table notifications.

Activate Live Feed

Live Feed is active for all instances. For instances upgrading from a version prior to Fuji, an administrator can activate the Live Feed v2 plugin.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Document feeds

A document feed is a live feed group that is associated with a record, such as an incident or change.

Document feeds allow users to work on tasks and other records through the live feed interface. Users can post messages in live feed that are automatically maintained as comments or work notes on the record, if the record has these journal fields.

The benefits of using document feeds include the following.

- Improving communication between users working on the same record.
- Improving visibility into progress for end users.
- Providing a single place to see updates on multiple records.
- Saving conversation history in the record so that knowledge is not lost in an email chain.

To use document feeds, activate the Live Feed Document plugin, which is active by default in new instances. See Use Live Feed to work on records on page 2379 to learn how to use this feature.

The system automatically creates a document group for the feed when a user follows or shows a record on live feed. The system also creates a document group when a user creates a new record on a table that has live feed enabled.

The following list describes features of the document feed group.

- Is unlisted; it does not appear when users view all groups on live feed.
- Automatically approves membership for every user who can access the record.
- Uses the record number as the group name.
- Uses the record short description as a group description.
- Maintains all messages posted to live feed on the record, if the record has a journal field for comments.
  If the record has a standard text field for comments, each live post overwrites the field value.
• Maintains all messages posted on the record in live feed if the record has a journal field for comments. When the group is created, existing messages are added to the document feed.

A user who participates in live feed becomes a member of the group.

*Add a live feed UI action on a table*
You can add UI actions on a table to allow users to follow records in live feed.

Role required: `ui_action_admin` or `admin`

1. Navigate to **System Definition > UI Actions**.
2. Open one of the live feed UI actions, for example, the Follow on Live Feed list UI action.
3. In the `Table` field, select the table name.
4. Right-click the header and select **Insert** to create a copy of the UI action for the desired table.
5. Repeat steps 1 – 4 for all live feed UI actions.

*Add the Show Live Feed button in the form header*
You can add the Show Live Feed button in the form header for a table.

Role required: `personalize_dictionary` or `admin`

1. Navigate to **System Definition > Tables**.
2. Open the table record.
3. Select the **Live feed** check box.
4. Click **Update**.

If the check box is selected and the live feed functionality is not active, open the dictionary entry for the table and add `live_feed=true` in the `Attributes` field. Click **Update**.

*Security configuration for document feeds*
Document feeds honor the access control rules (ACLs) for the associated record. Users can only view messages on the document feed if they have access to the same information on the record.

Consider the following examples.

• If an ACL allows a user to read and create comments on an incident, then the user can view and add messages posted as comments on the incident feed.

• If an ACL restricts a user from reading work notes, then the user cannot view messages posted as work notes on the incident feed.

**Note**: Access control rules are only checked when a user first accesses the document feed. After users view the feed, an administrator must remove them manually to change their access.

*Disable a record feed*
You can disable live feed functionality from the form of any table.

Role required: `personalize_dictionary` or `admin`

1. Navigate to **System Definition > Dictionary**.
2. Open the dictionary entry for the table.
3. Add `live_feed=false` in the `Attributes` field.
4. Click **Update**.
Note: If the Collaboration feature is activated, you can remove the show live feed icon ( ) from all form headers. Set the glide.live_feed.task_header_button property to collaboration.

Disable a document feed
You can disable a document feed from the form of any table.

Role required: personalize_dictionary or admin

1. Navigate to System Definition > Tables.
2. Open the table record.
3. Clear the Live feed check box.
4. Click Update.

Business rule installed with live feed document
This business rule is installed with Live Feed Document. There are no tables, roles, or notifications installed with it.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live Feed integration</td>
<td>Writes journal comments to the live feed if there is a group for this record.</td>
</tr>
</tbody>
</table>

Record feeds
A record feed is a live feed group that is associated with a record, such as an incident or change.

Record feeds allow users to work on tasks and other records through the live feed interface. Users can post messages in live feed that are automatically maintained as comments or work notes on the record if the record has these journal fields.

You can use record feeds to:
- Improve communication between users working on the same record.
- Improve visibility into progress for end users.
- Provide a single place to see updates on multiple records.
- Save feed history in the record so that knowledge is not lost in an email chain.
Activate live feed document
The Live Feed Document plugin is active by default. Administrators can enable record feeds in an instance by activating the Live Feed Document plugin, which activates the Live Feed plugin, if it is not active. For upgrades, administrators must activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Configure record feeds
Configure a table to support record feeds. You can add the Show Live Feed button in the form header and add the UI actions Follow on Live Feed and Show Live Feed as List and Form.

By default, the Show Live Feed icon appears on the form header for all tables that extend task, including the Incident, Change Request, and Problem tables, and in the list and form context menus for all task tables.
Add live feed to a form header

Add the **Show Live Feed** button in the form header for a table.

1. Navigate to **System Definition > Tables**.
2. Open the table record.
3. Select the **Live feed** check box.
4. Click **Update**.
   Alternatively, you can add `live_feed=true` to the **Attributes** field in the dictionary entry for the table.

Add live feed to a context menu

Add live feed UI actions on a table.

1. Navigate to **System Definition > UI Actions**.
2. Open one of the live feed UI actions, for example, **Follow on Live Feed**.
3. In the **Table** field, select the table name.
4. Right-click the header and select **Insert** to create a copy of the UI action for the desired table.
5. Repeat steps 1 – 4 for all live feed UI actions.

Add a live feed UI action on a table

You can add UI actions on a table to allow users to follow records in live feed.

Role required: `ui_action_admin` or `admin`

1. Navigate to **System Definition > UI Actions**.
2. Open one of the live feed UI actions, for example, the **Follow on Live Feed** list UI action.
3. In the **Table** field, select the table name.
4. Right-click the header and select **Insert** to create a copy of the UI action for the desired table.
5. Repeat steps 1 – 4 for all live feed UI actions.

**Disable a record feed**

You can disable live feed functionality from the form of any table.

Role required: `personalize_dictionary` or `admin`

1. Navigate to **System Definition > Dictionary**.
2. Open the dictionary entry for the table.
3. Add `live_feed=false` in the **Attributes** field.
4. Click **Update**.

**Note:** If the Collaboration feature is activated, you can remove the show live feed icon (ー) from all form headers. Set the `glide.live_feed.task_header_button` property to `collaboration`.

**Configure record feed security**

Record feeds honor the access control rules (ACLs) for the associated record.

Users can only view messages on the record feed if they have access to the same information on the record. For example:

- If an ACL allows a user to read and create comments on an incident, then the user can view and add messages posted as comments on the incident feed.
• If an ACL restricts a user from reading work notes, then the user cannot view messages posted as work notes on the incident feed.

**Note:** Access control rules are only checked when a user first accesses the record feed. After users view the feed, an administrator must remove them manually to change their access.

Live feed functionality is not active for the table

What to do if the check box is already selected and the live feed functionality is not active for the table.

1. Navigate to System Definition > Dictionary.
2. Open the dictionary entry for the table.
3. Add `live_feed=true` in the Attributes field.
4. Click Update.

Live feed functionality still exists

What to do if the check box is already cleared and the live feed functionality still exists for the table.

1. Navigate to System Definition > Dictionary.
2. Open the dictionary entry for the table.
3. Add `live_feed=false` in the Attributes field.
4. Click Update.

User preferences available in Live Feed

The following preferences are available in live feed.

**Table 564: User preferences available in live feed**

<table>
<thead>
<tr>
<th>User Preference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>live_tag.feed.order</td>
<td>Tracks user's choice of tag display sorting.</td>
</tr>
<tr>
<td>live_message.feed.last_display</td>
<td>Tracks last feed user has viewed. The last viewed feed is displayed the next time the user opens live feed. The first time a user opens live feed, the Company Feed is displayed.</td>
</tr>
</tbody>
</table>

Components installed with Live Feed

Several types of components are installed with Live Feed.

Demo data is available with Live Feed.

Plugin installed with Live Feed

The Live Feed application uses the following plugin:
Table 565: Plugins

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Description</th>
</tr>
</thead>
</table>

Tables installed with Live Feed

The Live Feed plugin installs the following tables.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link attachments [live_link]</td>
<td>Stores links and attachments that are added to messages.</td>
</tr>
<tr>
<td>Live Favorite [live_favorite]</td>
<td>Stores the favorite conversations, user groups, and messages.</td>
</tr>
<tr>
<td>Live Follows [live_follow]</td>
<td>Maintains users’ follow preferences for user feeds.</td>
</tr>
<tr>
<td>Live Group Profile [live_group_profile]</td>
<td>Stores conversation properties, including name, description, and public status. Also stored are user group properties, such as name, description, and public status.</td>
</tr>
<tr>
<td>Live Group Member [live_group_member]</td>
<td>Maintains the member lists for conversation and user group.</td>
</tr>
<tr>
<td>Live Mention [live_mention]</td>
<td>Stores the profiles mentioned in a message. References sys_user table.</td>
</tr>
<tr>
<td>Live Message Tag [live_message_tag]</td>
<td>Stores messages associated with user-created tags. Appears as a related list on Live Tag records.</td>
</tr>
<tr>
<td>Live Poll [live_poll]</td>
<td>Stores a poll question.</td>
</tr>
<tr>
<td>Live Poll Cast [live_poll_cast]</td>
<td>Stores profiles of users who voted for a particular option in a poll.</td>
</tr>
<tr>
<td>Live Poll Options [live_poll_option]</td>
<td>Stores poll options.</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Live Table Notification</td>
<td>Generates automatic table notifications, such as when a user reads live feed for the first time and when a high priority incident is opened. Access at Collaborate &gt; Feed Administration &gt; Table Notifications.</td>
</tr>
<tr>
<td>Live Tag</td>
<td>Maintains user-created tags. Access at Collaborate &gt; Feed Administration &gt; Tags.</td>
</tr>
<tr>
<td>Live Tag Follows</td>
<td>Maintains users' follow preferences for tags.</td>
</tr>
<tr>
<td>Messages Liked by</td>
<td>Maintains like ratings for posts. Access at Collaborate &gt; Feed Administration &gt; Likes.</td>
</tr>
</tbody>
</table>

Properties installed with Live Feed

The Live Feed plugin installs the following properties.

To access live feed properties, navigate to Collaborate > Feed Administration > Properties.

Table 566: Properties installed with live feed

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lide.live_feed.max_popular_tags</td>
<td>Indicates the maximum number of popular tags displayed.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 36</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.live_feed.max_recent_tags</td>
<td>Indicates the number of recent tags to show in the tag display.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 36</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.live_feed.my_feed_enabled</td>
<td>Enables user access to My Feed capabilities.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.live_feed.company_feed_exclude_groups</td>
<td>Prevents messages that belong to a group from appearing on the Company Feed</td>
</tr>
<tr>
<td></td>
<td>To configure this property: navigate to the Collaborate &gt; Feed Administration &gt; Properties page and modify the Do not show messages that belong to a group on the Company Feed option.</td>
</tr>
<tr>
<td></td>
<td>To configure this property: modify glide.live_feed.company_feed_exclude_groups on the System Properties [sys_properties] table and set it to true or false.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.live_feed.message_poll_interval</td>
<td>Indicates how frequently live feed is updated with new messages.</td>
</tr>
<tr>
<td></td>
<td>To configure this property: navigate to Collaborate &gt; Feed Administration &gt; Properties. Enter the polling value in the Live message poll interval time in ms. field.</td>
</tr>
<tr>
<td></td>
<td>To configure this property: add the property glide.live_feed.message_poll_interval (it is not on the sys_properties table by default), and enter a value in milliseconds.</td>
</tr>
<tr>
<td></td>
<td>If you do not add the property, the default value poll interval time is 15000 (15 seconds). To minimize performance impact, the minimum value is 2000 (2 seconds).</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 15000</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.ui.show_live_feed_activity</td>
<td>Displays two tabs on a record form's activity formatter: Live Feed, which allows users to collaborate on the document feed, and Activity, which shows a summary of all activity for the record. This property is not enabled by default.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>
The Live Feed plugin installs the following user roles.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>live_feed_admin</td>
<td>Can manage live feed functions.</td>
</tr>
<tr>
<td>chat_admin</td>
<td>Can manage chat functions (if the Chat plugin is activated)</td>
</tr>
</tbody>
</table>

The Live Feed plugin installs the following script includes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiveMsgUtil</td>
<td>Provides helper functions for working with live_message records.</td>
</tr>
<tr>
<td>LiveFeedFilter</td>
<td>Queries filters for working with live_feed records.</td>
</tr>
<tr>
<td>LiveFeedUtil</td>
<td>Provides helper functions for working with live_feed records.</td>
</tr>
</tbody>
</table>

The Live Feed plugin installs the following script actions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 567: Script actions installed with live feed

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live Feed Update</td>
<td>Generates system messages when an update occurs on a table that is relevant to live feed.</td>
</tr>
</tbody>
</table>

Scripted web services installed with Live Feed

The Live Feed plugin installs the following scripted web services.

Table 568: Scripted web services installed with live feed

<table>
<thead>
<tr>
<th>Scripted Web Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostToLiveFeed</td>
<td>Performs the following functions:</td>
</tr>
<tr>
<td></td>
<td>• Defines the user or profile to use; otherwise, the message posts from the authenticated soap user.</td>
</tr>
<tr>
<td></td>
<td>• Defines the group (using the groupid); otherwise the message posts to the company feed.</td>
</tr>
<tr>
<td></td>
<td>• Includes a link in the post.</td>
</tr>
</tbody>
</table>

Sample SOAP envelope

```xml
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:m="http://www.service-now.com"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:execute>
      <message xsi:type="xsd:string">Post this message to the Live Feed</message>
      <!-- optional sys_id of a sys_user record to use to get a live profile -->
      <user xsi:type="xsd:string"></user>
      <!-- optional sys_id of a live profile to post as -->
      <profile xsi:type="xsd:string"></profile>
      <!-- optional sys_id of a live group to post the message in-->
      <group xsi:type="xsd:string"></group>
      <!-- optional URL of a link to include in the message-->
      <linkurl xsi:type="xsd:string"></linkurl>
      <!-- optional name for the linkurl parameter--> 
      <linkname xsi:type="xsd:string"></linkname>
    </m:execute>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Business rules installed with Live Feed

The Live Feed plugin installs the following business rules.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feed Group Creator Becomes Admin</td>
<td>Live Group Profile [live_group_profile]</td>
<td>Designates the group creator as the group administrator.</td>
</tr>
<tr>
<td>live feed events</td>
<td>Task [task]</td>
<td>Runs on task insert, update, and delete. Triggers event associated with the Live Feed Update Script action that processes Live Table Notifications to auto-generate live feed messages.</td>
</tr>
<tr>
<td>Live feed member update events</td>
<td>Live Group Member [live_group_member]</td>
<td>Generates a notification event when member state changes (invited, accepted, declined, left, rejected, request, request_accepted).</td>
</tr>
<tr>
<td>Live feed integration</td>
<td>Journal Entry [sys_journal_field]</td>
<td>Writes journal comments to the live feed if there is a group for this record.</td>
</tr>
<tr>
<td>Live feed new member events</td>
<td>Live Group Member [live_group_member]</td>
<td>Generates a notification event when new members are added.</td>
</tr>
<tr>
<td>live feed profile events</td>
<td>Live Profile [live_profile]</td>
<td>Runs on live_profile insert/update/delete, triggers event associated with the Live Feed Update script action that processes Live Table Notifications to auto-generate live feed messages.</td>
</tr>
<tr>
<td>Live Feed message events</td>
<td>Live Feed Message [live_message]</td>
<td>Runs on live_message, notification event trigger for new live messages.</td>
</tr>
<tr>
<td>Live message like events</td>
<td>Message Liked by [live_message_like]</td>
<td>Runs on live_message_like, notification event trigger for new like records.</td>
</tr>
<tr>
<td>LiveFeed Group Member Visibility 2.0</td>
<td>Live Group Member [live_group_member]</td>
<td>Ensures users can only see the members list for public groups and groups they belong to.</td>
</tr>
<tr>
<td>LiveFeed Group Profile Validation</td>
<td>Live Group Profile [live_group_profile]</td>
<td>Ensures that a public group is visible.</td>
</tr>
<tr>
<td>LiveFeed Group Profile Visibility 2.0</td>
<td>Live Group Profile [live_group_profile]</td>
<td>Ensures that the list of all groups only displays public groups, private groups that are visible, and groups the user belongs to.</td>
</tr>
<tr>
<td>Business rule</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LiveFeed Membership Changes</td>
<td>Live Group Member [live_group_member]</td>
<td>Enforces that only the group administrator and users with live_feed_admin role can manage membership for a group.</td>
</tr>
<tr>
<td>LiveFeed Single Group Membership</td>
<td>Live Group Member [live_group_member]</td>
<td>Ensures that a user is not added multiple times to the same group.</td>
</tr>
<tr>
<td>Live Message Likes</td>
<td>Live Message Like [live_message_like]</td>
<td>Updates the number of likes for a message.</td>
</tr>
<tr>
<td>LiveFeed Join Group Check</td>
<td>Live Group Member [live_group_member]</td>
<td>Ensures that users can not automatically join private visible groups.</td>
</tr>
<tr>
<td>Update Follow/Follower Counts</td>
<td>Live Follow [live_follow]</td>
<td>Updates the following/followers counts.</td>
</tr>
<tr>
<td>Live Feed Group</td>
<td>Assessable Record [asmt_assessable_record]</td>
<td>Creates/Deletes a live feed group for an assessable record</td>
</tr>
<tr>
<td>Live Feed Message Visibility</td>
<td>Live Feed Message [live_message]</td>
<td>Ensures user's access to live feed messages</td>
</tr>
</tbody>
</table>

**Events registered by Live Feed**

The following events are registered by Live Feed.

Table 569: Events registered by Live Feed

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>live_feed.feed_member.left</td>
<td>Triggers an email notification to administrators when a user has left a feed.</td>
</tr>
<tr>
<td>live_feed.feed_member.request_accepted</td>
<td>Triggers an email notification to a user when the user's request to join a feed has been accepted.</td>
</tr>
<tr>
<td>live_feed.feed_member.accepted</td>
<td>Triggers an email notification when a user accepts an invitation to join a feed.</td>
</tr>
<tr>
<td>live_feed.feed_member.request</td>
<td>Triggers an email notification to administrators when a user requests to join a feed.</td>
</tr>
<tr>
<td>live_feed.feed_member.declined</td>
<td>Triggers an email notification when an administrator has declined a user's request to join a feed.</td>
</tr>
<tr>
<td>live_feed.feed_member.removed</td>
<td>Triggers an email notification when an administrator has removed a user from a feed.</td>
</tr>
<tr>
<td>live_feed.feed_member.invited</td>
<td>Triggers an email notification when an administrator has invited a user to join a feed.</td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>live_feed.feed_member.rejected</td>
<td>Triggers an email notification to a user when the user's request to join a feed has been rejected by an administrator.</td>
</tr>
<tr>
<td>live_message.mentioned</td>
<td>Triggers an email notification when a user is mentioned in a message.</td>
</tr>
<tr>
<td>live_message_like.inserted</td>
<td>Triggers an email notification when another user likes a message that a user posts.</td>
</tr>
<tr>
<td>live_message.replied</td>
<td>Triggers an email notification to all users in a thread when a user posts a reply.</td>
</tr>
<tr>
<td>live_message.inserted</td>
<td>Triggers an email notification to subscribed users when a new message is posted.</td>
</tr>
<tr>
<td>live_message.group_inserted</td>
<td>Triggers an email notification to subscribed users when a new message is posted to a feed.</td>
</tr>
<tr>
<td>live_feed.team_member.invited</td>
<td>Triggers an email notification when a user is invited to join a team.</td>
</tr>
<tr>
<td>live_feed.team_member.accepted</td>
<td>Triggers an email notification when an user accepted to join a team.</td>
</tr>
<tr>
<td>live_feed.team_member.declined</td>
<td>Triggers an email notification when a user declines an invitation to join a team.</td>
</tr>
<tr>
<td>live_feed.team_member.request</td>
<td>Triggers an email notification to administrators when a user requests to join a team.</td>
</tr>
<tr>
<td>live_feed.team_member.request_accepted</td>
<td>Triggers an email notification when an administrator has accepted a user's request to join a team.</td>
</tr>
<tr>
<td>live_feed.team_member.rejected</td>
<td>Triggers an email notification to a user when the user's request to join a team is rejected.</td>
</tr>
<tr>
<td>live_feed.team_member.left</td>
<td>Triggers an email notification when a user leaves a team.</td>
</tr>
<tr>
<td>live_feed.team_member.removed</td>
<td>Triggers an email notification when a user has been removed from a team.</td>
</tr>
<tr>
<td>live_message.group_replied</td>
<td>Triggers an email notification to subscribed users when a reply to a message in a feed has been posted.</td>
</tr>
<tr>
<td>live_feed.update</td>
<td>Triggers the Live Feed Update script action.</td>
</tr>
</tbody>
</table>

**Email notifications installed with Live Feed**

The Live Feed plugin installs the following email notifications..

<table>
<thead>
<tr>
<th>Email notification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live Feed Feed Member Accepted</td>
<td>Sends an email to the administrator when a user has accepted to join the feed.</td>
</tr>
<tr>
<td>Live Feed Feed Member Declined</td>
<td>Sends an email to the administrator when a user has declined to join the feed.</td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Live Feed Feed Member Invited</td>
<td>Sends an email to a user when they have been invited to join a feed.</td>
</tr>
<tr>
<td>Live Feed Feed Member Left</td>
<td>Sends an email to the administrators when a user has left the feed.</td>
</tr>
<tr>
<td>Live Feed Feed Member Removed</td>
<td>Sends an email to user when they have been removed from the feed.</td>
</tr>
<tr>
<td>Live Feed Feed Member Request</td>
<td>Sends an email to administrators when an user requests to join a feed.</td>
</tr>
<tr>
<td>Live Feed Feed Member Request Accepted</td>
<td>Sends an email to the user when their request to join a feed is accepted.</td>
</tr>
<tr>
<td>Live Feed Feed Request Rejected</td>
<td>Sends an email to the user when their request to join a feed has been rejected by an administrator.</td>
</tr>
<tr>
<td>Live Feed MsgReply Subscription</td>
<td>Sends an email when a reply or reply_to_reply message is inserted into the thread of a message in a feed.</td>
</tr>
<tr>
<td>Live Feed Team Member Accepted</td>
<td>Sends an email to administrators when a user accepts an invitation to join a team.</td>
</tr>
<tr>
<td>Live Feed Team Member Declined</td>
<td>Sends an email to administrators when a user declines an invitation to join a team.</td>
</tr>
<tr>
<td>Live Feed Team Member Invited</td>
<td>Sends an email when a user is invited to join a team.</td>
</tr>
<tr>
<td>Live Feed Team Member Left</td>
<td>Sends an email when a user leaves a team.</td>
</tr>
<tr>
<td>Live Feed Team Member Removed</td>
<td>Sends an email when an administrator has removed a user from a team.</td>
</tr>
<tr>
<td>Live Feed Team Member Request</td>
<td>Sends an email when a user requests to join a team.</td>
</tr>
<tr>
<td>Live Feed Team Member Request Accepted</td>
<td>Sends an email when a user's request to join a team is accepted.</td>
</tr>
<tr>
<td>Live Team Member Request Rejected</td>
<td>Sends an email when a user's request to join a team is rejected.</td>
</tr>
<tr>
<td>Live Message All Subscription</td>
<td>Sends an email to subscribed users when any message (new or reply) is posted.</td>
</tr>
<tr>
<td>Live Message Feed Subscription</td>
<td>Sends an email when a new message is inserted into a feed.</td>
</tr>
<tr>
<td>Live Message Liked</td>
<td>Sends an email to the creator of a message when another user likes the message.</td>
</tr>
<tr>
<td>Live Message Mention</td>
<td>Sends an email to a user when that user is mentioned in a message.</td>
</tr>
<tr>
<td>Live Message New Posts Subscription</td>
<td>Sends an email to subscribed users when a new (not reply) message is posted.</td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Live Message Reply</td>
<td>Sends an email to all users in a feed thread when a user posts a reply (live_message.replied event).</td>
</tr>
</tbody>
</table>

Security rules (ACL) installed with Live Feed

The Live Feed plugin installs the following security rules on existing tables.

**Table 570: Security rules installed with live feed**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys_attachment</td>
<td>Allows users to read and write live feed profile attachments.</td>
</tr>
</tbody>
</table>

Live Feed security

Administrators and users with the live_feed_admin role can configure live feed security.

Live Feed security and table access

By default, administrators and users with the live_feed_admin role can configure live feed security and view all live feed tables.

You can also modify the following live feed tables.

- Live Table Notification [live_table_notification]: set up automatic messages that are generated when specific records are updated throughout the system.
- Live Feed Message [live_message]: modify only if necessary, such as to delete an inappropriate message or restore a message that a user accidentally deleted. This table is typically managed by system functionality.

Extending or modifying data in any live feed chat table is not recommended.

*Domain separation in Live Feed*

Using domain separation with the Live Feed application allows administrators to keep user content within a specific domain.

Parent and child domains define the live feed content that can be viewed and shared with other users.

- A user in a parent domain can see users and content within their domain and within all child domains that are lower in the domain hierarchy.
- A user in a child domain can see users and content within their domain but cannot see the parent domain or other child domains at the same level in the domain hierarchy.

Enable domain separation for live feed

Domain separation for the Live Feed application is available starting with the Eureka release. For users upgrading to Eureka, a sys_domain column is added to existing live feed-related database tables. For existing database records, the value of the sys_domain field is set to empty (global). This allows the domain separation feature to work with existing live feed records.
Show or hide the live feed application for a domain

The administrator can live feed visibility for a domain. This includes:

• Displaying live feed in the application navigator.
• Following a document feed for a record.
• Viewing live feed from a record.

Use live feed with domain separation

All of the live feed features are available to users within a domain and work the same way, with some exceptions to visibility.

• Posting and sharing content
  • Users in a child domain can interact with other users in the same domain. This includes posting and replying to messages, deleting messages, attaching files and links, and rating content.
  • Users in a parent domain can interact with other users in the same domain and with users in any child domains. Messages that are added to child threads by parent users are given the visibility of the child domain, not the parent domain.

• Viewing content
  • Users in a child domain can view content that resides within their domain. This includes filtering by feed, sorting, searching, and viewing older messages.
  • Users in a parent domain can view content that resides within their domain and within any child domains.
  • My Feed shows messages, teams, and hashtags based on the user’s domain. Showing another user’s feed only shows posts visible in the current user’s domain.

• Using hashtags
  • Hashtags are separated by domain. Users in a child domain can use all of the hashtag functions within their domain. This includes tagging messages, viewing available hashtags, changing hashtag names, and searching and filtering by hashtag.
  • Users in a parent domain can use all of the hashtag functions within their domain and can see hashtags in any child domains.

• Using teams
  • Teams are separated across domains. Users in a child domain can join and follow teams within their domain, and can invite other users within their domain to join teams.
  • Users in a parent domain can join and follow teams that belong to their domain or to any child domains.

• Exceptions to teams
  • New teams are created in the user’s domain. However, if the team has a record associated with it, the team is created in record’s domain.
  • New threads in an existing team are created in the domain of the existing team.
  • When a user is creating a new team from a document, if the document domain is not empty or global, the team domain needs to be changed to be the document’s domain.
  • For a document-generated team, when the domain of the document changes, the domain of all related live feed records changes as well.

• Subscribing to email notifications
- Users can subscribe to email notifications from users and teams within their domain only.

*Limit Live Feed access by role*
All active users in the instance have access to Live Feed by default.
Role required: live_feed_admin, admin

1. Complete the following steps to define the roles that allow users to see the Live Feed module.
   a) Perform the appropriate action for your version of the UI:
      - UI16: Navigate to System Definition > Application Menus > Collaborate and select Live Feed.
      - UI15 or UI11: Right-click the icon beside the Live Feed module and select Edit Module.
   b) Enter the roles that have access in the Roles field.
   c) Click Update.

2. Complete the following steps to define the roles that allow users to see live feed from a mobile device.
   a) Navigate to System Definition > Applications (Mobile).
   b) Click Live Feed.
   c) Enter the roles that have access in the Roles field.
   d) Click Update.

3. Use the following settings in the Access Control form to create access control rules to limit who can view the live_feed pages.
   - **Type**: ui page
   - **Operation**: read
   - **Name**: create one access control rule record for each of the following pages
     - live_feed
     - live_feed_small
     - $live_feed
     - $live_feed_small
   - **Requires role**: in this related list, add roles to define who can access the live feed pages. Users who do not have these roles cannot access live feed.

*Manage Live Feed message content*
Users can remove their own messages from feeds. If necessary, administrators can remove inappropriate messages that are posted by any user.
Role required: live_feed_admin, admin

For security reasons, any HTML code is automatically stripped out of a message before it is posted. This measure ensures that users cannot modify any page settings by posting a message.

1. Navigate to Collaborate > Feed Administration > Messages.
2. Open the message to be removed.
3. In the State field, select deleted. You may need to configure the form to add this field.
4. Click Update.

*Live Feed team security*
You can restrict who can create teams by modifying an access control rule.
Modify the following access control rule.

- **Table**: Live Group Profile [live_group_profile]
- **Operation**: create

*Restrict hashtag renaming in Live Feed*

You can restrict who can rename hashtags.

You restrict hashtag rename by modifying the following access control rule:

- **Table**: Live Tag [live_tag]
- **Operation**: write

*Provide access to Live Feed from CMS pages*

You can provide access to live feed from pages built in the Content Management System (CMS). For example, allow an end user to access your company feed via the ESS portal.

The ESS Portal template includes the Portal - Live page and Live Feed dynamic blocks. To provide access to live feed from CMS pages, add the Live Feed dynamic block to a CMS page or include Portal - Live page in a site.

To add live feed in an iFrame:

1. Navigate to **Content Management > iFrames**.
2. Click **New**.
3. Enter the iFrame block details.

### Table 571: iFrame block fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name such as Live Frame.</td>
</tr>
<tr>
<td>Frame Name</td>
<td>Enter a frame name, such as live_frame.</td>
</tr>
</tbody>
</table>
| URL            | https://INSTANCE/live_feed.do? sysparm_doctype=true  
|                | where INSTANCE is your instance URL (example, [instance name].service-now.com) |
| Application    | Displays scoping information.                                               |
| Sizing         | Select **Fixed Size** and enter height and width pixel dimensions according to the page where you plan to display the feed (for example, width of 1024 and height of 768). |

4. Click **Submit**.
5. Add the block to a page.

Live Feed table notifications

Live feed table notifications generate automatic live feed messages. The Live Feed plugin must be activated to use table notifications.

When a record is inserted or updated on a specific task table and the notification conditions are met, a message is generated and posted to the specified group or to the company feed. Table notifications are supported for all task tables and all task types by default.

Administrators and users with the live_feed_admin or chat_admin roles can set up table notifications for any table in an instance.

Table notifications used by Live Feed

The following table notifications are used by Live Feed.

<table>
<thead>
<tr>
<th>Table notification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User joined live feed</td>
<td>Active by default.</td>
</tr>
<tr>
<td>High priority incident opened</td>
<td>Not active by default.</td>
</tr>
</tbody>
</table>
Set up table notifications for task tables

You can set up a table notification for a task table to enable automatic live feed messages.

Role required: live_feed_admin or chat_admin

1. Navigate to Collaborate > Feed Administration > Table Notifications.
2. Click New or select a notification to open it.
3. Complete the form.

Table 572: Live Table Notification fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Select the table for which notifications are generated.</td>
</tr>
<tr>
<td>Post to chat rooms</td>
<td>Select the check box to generate the notification in chat rooms that are associated with the record. This applies to task records only and requires that the Chat plugin is active.</td>
</tr>
<tr>
<td>Post to live feed</td>
<td>Select the check box to generate the notification on live feed. If record feeds are not set up and the Feed group field is blank, the notification is posted to the company feed.</td>
</tr>
<tr>
<td>Record Feeds</td>
<td>Select the fields to post for record feeds. This field is available only if Post to live feed is selected.</td>
</tr>
<tr>
<td>Application</td>
<td>Select the application that contains this record.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to enable the notification.</td>
</tr>
<tr>
<td>Insert</td>
<td>Select the check box to generate the notification when a record is inserted into the database.</td>
</tr>
<tr>
<td>Update</td>
<td>Select the check box to generate the notification when a record is updated.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Feed</td>
<td>Select the live feed team to which you want to post messages. This field is available only if <strong>Post to live feed</strong> is selected. Clear the field to post notifications on the company feed.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Create the condition that must be met to generate the notification. If you add a condition statement, the system evaluates the condition first and parses the <strong>Before script</strong> field only if the condition is met. You may choose to leave this field blank and include conditions in the script.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description of the table notification.</td>
</tr>
</tbody>
</table>
| Message    | Construct the automatic message you want to generate. You can include field values as variables and a link to the record. To add a variable, position the cursor in the text and then click the field name in the **Select variables** field. To add a link to the record, enter `${URI}` in the message text as shown in the following example. 

```
High priority incident opened: 
${URI}
Short description: 
${short_description}
```

**Note:** For security reasons, any HTML code is automatically stripped out of a message before it is posted.
4. Click Submit or Update.

Set up table notifications for non-task tables

Table notifications are supported for all task tables by default, and you can set up a table notification for any other table.

Role required: live_feed_admin or chat_admin

1. Navigate to System Definition > Business Rules.
2. Open the live feed events business rule.
3. In the Advanced section, select and copy the text in the Script field.
4. Click the back arrow on the top, left to exit the record and return to the Business Rules list.
5. Click New.
6. Enter the following values at the top of the form.
   • Name: Enter a name, such as live_feed events for my table.
   • Table: Select the table for which you are setting up a notification.
   • Active and Advanced: Select these check boxes.
7. In the When to run section, enter the following values.
   • When: Select before.
   • Insert, Update, and Delete: Select the check boxes.
8. In the Advanced section, paste the script from the live feed events business rule into the script box.
9. Click Submit.
The live feed events business rule runs on the non-task table. It fires the live_feed.update event, which is associated with the Live Feed Update script action. The script action sets up variables and processes the table notifications.

After saving the business rule, create a live feed table notification record for the new table as you would for a task table.

Live feed table notification examples

The following examples illustrate different scenarios of table notification usage in live feed.

**Example 1: workaround posted**

This example demonstrates a table notification that generates an automatic message on live feed whenever a workaround is added to an open problem.

- **Table**: Problem [problem]
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions**: State is Open
- **Description**: Workaround Posted
- **Message**:

```plaintext
${sys_updated_by} posted a workaround for ${URI}.
Short description: ${short_description}
```

- **Before script**:

```plaintext
// only post to live feed when the Workaround field changes
answer = changedFields.contains("work_around");
```
Example 2: problem resolved (advanced)

This advanced example demonstrates a table notification that generates an automatic message on live feed whenever a problem is closed. It also adds a message about the assigned user and posts the message from the assignment group profile instead of the problem record.

- **Table**: Problem [problem]
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions**: Problem State is Closed/Resolved
- **Description**: Problem Resolved
- **Message**: Problem ${number} - ${short_description} has been resolved. ${fixedByMsg}

- **Before script**:

```java
//cancel if we didn't just change the problem state if ( !changedFields.contains ( "problem_state" ) )
answer = false ;
```
//if we have an assigned_to value add a comment about who it was
//create a new variable fixedByMsg that we can access from the message
fixedByMsg = "" ; if ( !current. assigned_to. nil () )
fixedByMsg = " Thank you " + current. assigned_to. getDisplayValue () ;

//make the message appear to come from the assignment group if we have one if ( !current. assignment_group. nil () )
profileSource = current. assignment_group. getRefRecord () ; //need GlideRecord object

---

Figure 667: Problem resolved (advanced) example
**Example 3: opportunity won (non-task)**

This example demonstrates a table notification on a non-task table. It generates an automatic message on live feed whenever a sales opportunity is won (requires the Sales Force Automation plugin).

Create a business rule on the Opportunity [sales_opportunity] table that matches the live feed events business rule on the task table.

Create a table notification with the following values:

- **Table**: Opportunity [sales_opportunity]
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions**: State is Closed Won
- **Description**: Opportunity won
- **Message**:

  ```
  ${owner} closed a sale with ${account}!
  ```

- **Before script**:

  ```
  //make the message appear to come from the assigned salesperson if there is one if (!current. owner. nil ( ) )
  profileSource = current. owner. getRefRecord ( ) ; //need GlideRecord object
  ```
**Example 4: table notification for related feeds**

This example demonstrates table notifications to be sent out to related feeds. For this example, a table notification is set up so that whenever the status of a story changes to Complete, a message is sent to the related sprint, release, or epic. Messages are posted only if the related feed already exists; this notification does not create a new feed.

Create a table notification with the following values:

- **Table**: Story [rm_story]
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Record feeds**: move Sprint, Release, and Epic to the **Selected** column.
- **Conditions**: State changes to Complete
- **Description**: Story is done; message to Epic, Release, and Sprint
- **Message**:

  ```
  ${URI} status changed to ${state}
  ```
Workaround posted
This example demonstrates a table notification that generates an automatic message on live feed whenever a workaround is added to an open problem.

- **Table**: Problem [problem]
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions**: [State] [is] [Open]
- **Description**: Workaround Posted
- **Message**:

  ```
  ${sys_updated_by} posted a workaround for ${URI}.
  Short description: ${short_description}
  ```

- **Before script**:

  ```
  // only post to live feed when the Workaround field changes
  answer = changedFields. contains ( "work_around" ) ;
  ```

Figure 669: Example workaround message

Problem resolved (advanced)
This advanced example demonstrates a table notification that generates an automatic message on live feed whenever a problem is closed.
It also adds a message about the assigned user and posts the message from the assignment group profile instead of the problem record.

- **Table:** Problem [problem]
- **Active:** Select the check box.
- **Update:** Select the check box.
- **Post to live feed:** Select the check box.
- **Conditions:** [Problem State] [is] [Closed/Resolved]
- **Description:** Problem Resolved
- **Message:**

  Problem ${number} - ${short_description} has been resolved. ${fixedByMsg}

- **Before script:**

  ```java
  //cancel if we didn't just change the problem state if ( !changedFields.
  contains ( "problem_state" ) )
  answer  = false ;
  //if we have an assigned_to value add a comment about who it was //create
  a new variable fixedByMsg that we can access from the message
  fixedByMsg  = "" ; if ( !current. assigned_to. nil ( ) )
  fixedByMsg  = " Thank you " + current. assigned_to. getDisplayValue ( ) ;
  //make the message appear to come from the assignment group if we have
  one if ( !current. assignment_group. nil ( ) )
  profileSource  = current. assignment_group. getRefRecord ( ) ; //need
  GlideRecord object
  ```
Figure 670: Example problem resolved message

**Opportunity won (non-task)**

This example demonstrates a table notification on a non-task table. It generates an automatic message on live feed whenever a sales opportunity is won, if you have activated the Sales Force Automation plugin.

Create a **business rule** on the Opportunity [sales_opportunity] table that matches the **live feed events** business rule on the task table.

Create a table notification with the following values:

- **Table**: Opportunity [sales_opportunity]
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions:** [State] [is] [Closed Won]
- **Description:** Opportunity won
- **Message:**

```
${owner} closed a sale with ${account}!
```

- **Before script:**

```
//make the message appear to come from the assigned salesperson if there is one
if (!current.owner.nil())
profileSource = current.owner.getRefRecord(); //need GlideRecord object
```

---

**Figure 671: Example opportunity won message**

*Related feeds table notification*

This example demonstrates table notifications to be sent out to related feeds.

For this example, whenever the status of a story changes to **Complete**, a table notification message is sent to the related sprint, release, or epic. Messages are posted only if the related feed already exists; this notification does not create a new feed.

- **Table:** Story [rm_story]
- **Active:** Select the check box.
- **Update:** Select the check box.
- **Post to live feed:** Select the check box.
- **Record feeds:** Move **Sprint**, **Release**, and **Epic** to the **Selected** column.
• **Conditions:** [State] [changes to] [Complete]
• **Description:** Story is done; message to Epic, Release, and Sprint
• **Message:**

${URI} status changed to ${state}

---

**Use Live Feed**

Live Feed provides many methods you can use to share content with others in your organization.

Depending on your role, there are several different types of feeds you can access, teams you can join and share information with, and hashtags you can use for categorizing messages.
**Note:** Live Feed v2 is active for all new instances by default. If you are upgrading from an earlier version of ServiceNow, you need to activate Live Feed v2 to use these features.
Live Feed browser support

The Live Feed v2 plugin does not support Internet Explorer 7 through 9.

Users who access the instance from those browser versions can use Live Feed v1 functionality. Users who access the instance from browsers that are compatible with HTML5 can use Live Feed v2 functionality.

Use hashtags in Live Feed

Hashtags are words marked with a hash symbol (#) in messages. Hashtags are a way to categorize messages by keyword or topic for improved search results.

For example, to identify messages associated with VPN questions, add the hashtag #VPN in the message text.

---

**Note:** Many of these features are part of live feed v2.

---

Tag messages

To tag a keyword or topic in a message, enter a # symbol before the word. A link to the hashtag is added beneath the message and all feed users can search and filter by the hashtag to find the message.

Tips for using hashtags effectively include:

- Limit the number of hashtags per message (no more than 3 is a good guideline).
- Use hashtags only in messages that are relevant to the tagged topic.
- Write a message first, and then add hashtags only if they add value. Answer the question, "What are you working on?", rather than "What hashtags apply to what you are working on?"
- Use camel case to create a hashtag that is more than one word long. For camel case, remove the space and use an uppercase letter to start each word. For example, to create a topic about the service desk, use #ServiceDesk.
Figure 673: Tag

View an available hashtag

1. Navigate to Collaborate > Live Feed.
2. Click Hashtags in the sidebar.
   Any hashtags already defined are displayed. Up to 36 hashtags can be displayed, and options are available for searching and sorting the hashtags. When searching, you do not need to include the hash symbol (#).
3. Click a hashtag to view all messages containing that hashtag.

Rename and merge hashtags

Over time, users may create many hashtags with similar names, such as competitive, competitor, and competition. To combine hashtags for better searching and filtering, change the names of similar hashtags to a standard name.

Administrators can restrict who can rename hashtags.

1. Navigate to **Collaborate > Live Feed**.
2. Click **Hashtags** in the sidebar to reveal the currently defined hashtags.
3. Turn off the filter to show all hashtags.
4. Point to the hashtag you want to change. If you have the rights to edit hashtags, an edit tag icon appears.

5. Click the edit tag icon.

<table>
<thead>
<tr>
<th>Hashtags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Competitive</td>
</tr>
<tr>
<td>2 Lync</td>
</tr>
<tr>
<td>1 MacBook</td>
</tr>
</tbody>
</table>

6. Enter a new hashtag name.
7. Press Enter.
   - If the new name is different than the other hashtags, the new hashtag name replaces the old hashtag name on the list and in the text of any messages that include the hashtag.
   - If the name is the same as one of the existing hashtags, the hashtags will be merged. For example, if you have two hashtags called Competitive and Competitors, and you change the name of Competitors to Competitive, all of the messages previously tagged Competitors are changed to Competitive and the Competitors hashtag is removed.

   **Note:** If a hashtag is removed from the message, renaming that hashtag does not change the message text.

Add an image to a hashtag

You can add an image at the domain level of already defined hashtags if you have the admin role.

1. Click Hashtags in the sidebar.
2. Select the appropriate hashtag.

   #VPN Feed -

3. Do one of the following:
   - Locate the image in your file system, and then drag-and-drop the image onto the add an image icon.
   - Click the add an image icon, select the image file, and click Open.

4. To change the image, simply drag-and-drop a new image onto the old one, or click on the existing one and select a new image.
When you view a message that contains a hashtag with an image, that image appears in the hashtag heading bar.

Follow and unfollow a hashtag

You can view all postings assigned a specific hashtag by following the hashtag. You can also unfollow the hashtag if you no longer want to view those postings.

1. Navigate to Collaborate > Live Feed.
2. Click Hashtags in the sidebar.
3. Select the hashtag you want to follow.
4. Click Follow.
A confirmation message appears.

5. Click X to dismiss the message.
   To unfollow a hashtag, repeat the same steps and click Unfollow.

Bookmark a hashtag

The Live Feed interface uses the standard bookmarking functionality for hashtags.

In UI15 and UI11, the live feed interface uses the standard bookmarking functionality for hashtags. You can drag-and-drop group feeds and hashtags from the Feed Summary section to the Edge for quick access.
Figure 674: Bookmark hashtag

Remove a hashtag from a posted message

As a participant in a conversation, you can remove hashtags from messages that have already been posted.

1. Navigate to the posted message.
2. Click Hashtags in the message posting.

The hashtags associated with the message are displayed, along with the removal icon.
3. Click the removal icon.

Use Live Feed to work on records

A record feed is associated with a record, such as an incident or change.

Record feeds help users collaborate on records by providing a place for anyone who can access the record to share messages and post attachments specific to that record.

With record feeds, users can:

• Follow record feeds and post messages in live feed. These messages can also be automatically maintained in the comments or work notes journal fields on records.
• View live feed from records.
• Work on multiple records from My Feed.
• Access live feed team functions, such as sending invitations and subscribing to email notifications.

Any users with access to the record can also use the record feed. By default, record feeds are available on the incident, change, and problem tables. Administrators can configure record feeds for additional tables.
How Document Feeds Work

The Live Feed application creates a document group for each document feed. The document group:

- Automatically approves membership for every user who can access the record.
- Uses the record number as the group name.
- Uses the record short description as a group description.
• Maintains all messages posted on the record in live feed (if the record has a journal field for comments). When the group is created, existing messages are added to the document feed.
• Lists the group when users select **View all groups** on their live feed interface, unless the record associated with the document feed has been closed. When the state of the record is closed, the live feed group becomes inactive and unlisted.
• Automatically adds users to the document group when they view the record.

**Document Group Creation**

When a user follows or shows a record on live feed, a live feed group is automatically created and associated to the record (if one does not already exist). The user becomes a member of the group and can use live feed to work on the record. If the user can access work notes on the record, the user also becomes a group administrator.

A live feed group is also automatically created when a user creates a record on a table that uses document feeds, such as the Incident table. The user who creates the record becomes the administrator of the group, and any other user who modifies the same record automatically joins the group.

**Control how live feed is enabled for records**

A property called glide.ui.show_live_feed_activity controls whether live feed is automatically enabled for all records.

This glide record is disabled by default. If the property is enabled, a record feed is automatically created whenever a record is viewed, and you become a member of that feed.

A second property, glide.live_feed.auto_join_document_group, enables automatic membership to a record feed whenever the associated document is visited and the user clicks **Show Live Feed**.

**View the activity stream in record feeds**

When you view record feeds, live feed displays the activity stream for the document.

For example, for an incident feed, the screen displays recent activity for the incident. The activity stream is updated whenever anything is changed in the associated record.
Figure 676: Live feed activity stream

Note: The information in the activity stream is subject to the same ACL rules as those for viewing the underlying document.
Viewing Live Feed from records

Interact with the record feed in any form that has live feed enabled.

The record feed appears in a pop-up window. The record feed also appears on the record form's activity formatter if the administrator enables the record feed system property (glide.ui.show_live_feed_activity).

To access a record feed from the form, do one of the following:

- Click the Show Live Feed button in the form header. This displays the live feed pop-up.
- Right-click the form header and select Show Live Feed. This also displays the live feed pop-up.
- Scroll down to the activity formatter area on the form. Click one of the following tabs:
  - **Live Feed**: click to show a text box to type in the feed.
  - **Activity**: click to show the activity summary. The activity filter determines the content in the activity summary.
If the activity formatter or the Live Feed and Activity tabs are not visible, administrators can do the following:

- Configure the form layout and add Activities (filtered) to the form. This adds the activity formatter.
- Personalize the form layout and add Activities (filtered) to the form. This adds the activity formatter.
- Go to System Definition > Tables, access the table associated with the record, and verify that the live_feed dictionary attribute is set to true on the form. This adds live feed to the activity formatter.
• Go to Collaborate > Feed Administration > Properties and enable the following property: **Toggle the display of the live feed tab in the activity formatter**

**Working with record feeds**

When a user follows a record, the user becomes a member of a private group for the record.

Team members can:

• View the record feed from live feed.
• View a list of group members (other users who are following the record).
• Invite another user to join the group. Only users who have access to the record have access to the record feed.
• Subscribe to email notifications.
• Leave the team to stop following the record.

**How record feeds work**

A record feed is associated with a record. The ServiceNow system automatically creates a record feed when a user follows or shows a record on live feed. The system also creates a record feed when a user creates a new record on a table that has Live Feed enabled.

The record feed:

• Is unlisted, it does not appear when users view all feeds on live feed.
• Automatically approves membership for every user who can access the record.
• Uses the record short description as the feed name, and the concatenation of the record number and record short description as the feed description. If the record short description is blank, then the record number is used as the feed name.
• Maintains all messages posted to live feed on the record, if the record has a journal field for comments. If the record has a standard text field for comments, each live post overwrites the field value.
• Maintains all messages posted on the record in live feed, if the record has a journal field for comments. When the feed is created, existing messages are added to the record feed.

**Follow a record feed from a form**

You can follow records that have a record feed associated with them from within the record form.

1. Navigate to the record form.
2. Click the **Show live feed** icon in the form header.
A Live Feed window is displayed.

3. Add a post to the record feed to start seeing the feed in your stream.

*Follow a record feed from a list*

You can follow records that have a record feed associated with them when viewing a list of records. This method allows you to select multiple records to follow.

1. Navigate to a list of records (for example, **Assigned to me**).
2. Select the **Action** check box beside each record to follow.
3. In the **Actions** choice list, select **Follow on Live Feed**.
The selected records appear in the record feeds and the activity stream.

**Post a message to a record feed**

1. Open the record feed in one of the following ways:
   - From your homepage (must already be following the record)
   - As a team in live feed (must already be following the record)
   - From My Live Feed (must already be following the record)
   - From the **Live Feed** pop-up window
   - From the **Live Feed** tab on the activity formatter
   - Click the **Show Live Feed** icon (●) in the form header.

2. Compose the message and add images and links.

3. [My Live Feed only] In the To list, select the record where you want to post the message.

4. [Optional] Select the **Work Notes** check box to post the message as a work note. Only users with access to work notes on the record can post and see work notes. This option is only available for records that have a journal field.

5. Click **Post**.

**Note:** If the record has journal fields for comments or work notes, your message is automatically added to the appropriate field on the record. If the record has a standard text field for comments on the activity formatter, each live post overwrites the field value. Comments added through live feed start with #LiveFeed on the activity formatter.
Use groups in Live Feed

Groups allow users to create focused discussions in live feed.

Users can view or join groups based on the access level of the group:

- **Public**: any user can see the name on the list of all groups, see the feed, and join. Messages appear on the company feed with a link to the group feed.
- **Private**: any user can see the name on the list of all groups, but only invited members can see the feed and join. Messages do not appear on the company feed.
- **Unlisted**: only invited users can see the name on the list of all groups, see the feed, and join. Messages do not appear on the company feed. Private groups may be listed or unlisted.

The group access level is configured by the user who creates the group. The group creator is also the group administrator. Any user can create a group; however, ServiceNow administrators can restrict the ability to create groups.

Automatic group creation

If you create records on tables that have live feed enabled, such as the Incident table, a live feed group is automatically created at the same time.

Automatic group participation

Users who view records on tables that have live feed enabled, such as the Incident table, automatically join the document group.

If the user creates a record on a table with live feed enabled, the user becomes the document group administrator. Document groups like these are not public. Any user who has access to the record also has access to the document group associated with that record.

Create a group (group administrators)

When you create a group, you become the group administrator.

Group administrators can:

- Modify group properties
- Accept or reject membership requests
- Delete the group
- Perform all group member functions

1. Navigate to **Collaborate > Live Feed**.
2. Click **Create Group**.
3. Enter a **Name** to appear at the top of the group feed and in messages posted to the group (links to the group feed).
4. Enter a **Description** to appear at the top of the group feed and under the group name in the list of all groups.
5. Configure the access level for the group:
   - For a public group, select **Public**.
   - For a private group, select **Private** and select the **Show this group when viewing all groups** check box.
   - For an unlisted group, select **Private** and clear the **Show this group when viewing all groups** check box.
6. Optional: Select a picture for the group.
   1. Click the picture beside the group name.
   2. Browse to the desired picture file and click Open.

7. Click Create. The group is created with you as the group administrator.

Delete a group

1. Navigate to Collaborate > Live Feed.
2. Click Groups (/groups) and select the group name.
3. Click Delete Group (/groups_delete).
4. Click Yes to confirm deletion. The group is deleted from live feed. Messages posted to a private group are deleted; messages posted to a public group remain on the company feed without a group link.

If you delete a group that is associated with an active record, such as an incident that has not yet been closed, the group becomes active again when any user modifies that record or follows the record on live feed. The first user who modifies the record becomes the group administrator. The previous messages that existed as part of the document feed are not deleted.

Invite new members

Any group member can invite another user to join a group.

1. Navigate to Collaborate > Live Feed.
2. Click Groups (groups) and select the group name.
3. Click Invite Member (invite_member).
4. Begin entering a user name and select a user from the list, or click the reference lookup icon (reference_lookup) and select a user from the table.
5. Click OK. The invited user receives an email notification.
Join a group (group members)

When you join a group, you become a group member.

Group members can:

• View messages posted to the group (group feed). Non-members can also view the feed for a public group.
• View a list of group members.
• Invite another user to join the group.
• Subscribe to group email notifications.
• Leave the group.

1. Navigate to Collaborate > Live Feed.
2. Click Groups ( ) and select View all groups.
   • If you received an invitation, click Accept. You can Decline the invitation if you do not want to join the group.
   • If the group is public, click Join.
   • If the group is private, click Request. The group administrator receives an email notification and must accept your request before you can join the group.

Leave a group

1. Navigate to Collaborate > Live Feed.
2. Click Groups ( ) and select the group name.
3. Click Leave.

Note:
If you leave a document group associated with a record, such as an incident, you automatically rejoin that group if you modify the associated record again.

Manage membership requests

When a user requests membership for a private group, the group administrator receives an email.

1. Navigate to Collaborate > Live Feed.
2. Click Groups ( ) and select the group name.
3. Click Members ( ).
4. In the Pending Members section, click Accept or Reject for each membership request.
   The user receives an email notification to confirm whether the request was accepted or rejected.
Modify a group

1. Navigate to Collaborate > Live Feed.
2. Click Groups ( ) and select the group name.
3. Click Modify Group ( ).
4. Modify the group properties or picture.
5. Click Update.

Subscribe to email notifications

1. Navigate to Collaborate > Live Feed.
2. Click Groups ( ) and select the group name.
3. Click Subscribe.

You can click Unsubscribe from a group to stop receiving email notifications.

Reply to email notifications using email

After receiving an email about a posted message or a reply, you can reply to live feed with your email client.

The body of the reply email becomes a reply to the live feed message. Any attachments included in your reply email are included in the live feed reply.

View a group feed

1. Navigate to Collaborate > Live Feed.
2. Click Groups ( ). A list of groups you belong to and an option to view all groups appear.
   - To view a group you belong to, select the group name.
   - To view a public group you do not belong to, select View all groups and then click the group name. You must be a member to view the feed for a private group.
View group members

1. Navigate to Collaborate > Live Feed.
2. Click Groups and select the group name.
3. Click Members. The list of group members appears, organized by group administrators, group members, and pending members.
Use feeds in Live Feed

Feeds allow users to create focused discussions in Live Feed. Live feed includes different types of feeds. The user who creates a group or record feed becomes the feed administrator. This user configures the access level for the feed:

- **Public**: any user can see the name on the list of all feeds, view the feed, and join it. Messages appear on the company feed with a link to the public feed.
- **Private**: any user can see the name on the list of all feeds, but only invited members can view the feed and join. Messages do not appear on the company feed.

**Note**: The concept of groups has changed in live feed v2 (starting with the Fuji release).

Create a group feed

Group feeds are created by users so that members with similar interests can find that information in one place. Individual users or teams can join group feeds.

1. Navigate to **Collaborate > Live Feed**.
2. Click **Group Feeds** and click **All Feeds**.
3. Click **Create Group Feed**.

4. Enter the feed name and description. These will appear in the list of group feeds.
5. Indicate how members can join the feed:
   - **Public Feed**: Select this to allow anyone to view and join the group feed.
   - **Private Feed**: Select this to restrict membership to only members invited by the administrator.

   When **Private Feed** is selected, the **Show this feed for anyone to find and request to join** check box is displayed. If it is selected, any user can search for this group feed and join it. If the check box is not selected, only users who are invited can see the feed and join the group feed.

6. Click **Create**.
Create a record feed

When a user views a record on a table that has live feed enabled, such as the Incident [incident] table, the user can click the **Show live feed** button at the top of the form to automatically create a record feed.

When a user who has write access to a record’s work_notes journal field clicks the **Show live feed** button, that user becomes the record feed administrator. If, for example, a customer calls technical support to report an incident and the customer clicks **Show live feed** on a record, a record feed is created, but the customer is not the record feed administrator. When a support representative later opens the incident and participates in the record feed, the user becomes the feed administrator.

Record feeds are, by default, private unlisted. That is, only invited members can see the name of the record feed in the list of feeds, and join it. Record feed administrators, however, have the ability to change the access level at any time. This can be helpful if, for example, a user is working on a sales opportunity and wants customers to be able to participate in a record feed without granting them access to the records.

Participate in a feed

When you join a feed, you become a feed member.

Feed members can:

- **View messages** posted to the feed. Non-members can also view the messages for a public feed.
- **View a list of feed members**.
- Invite another user to join the feed.
- Subscribe to feed email notifications.
- **Leave the feed**.

1. Navigate to **Collaborate > Live Feed**.
2. Click either **Group Feeds** or **Record Feeds**, and then click **All Feeds**.
• If you received an invitation, click **Accept**. You can **Decline** the invitation if you do not want to join the feed.
• If the feed is public, click **Join**.
• If the feed is private, click **Request**. The feed administrator receives an email notification and must accept your request before you can join the feed.

**Mark a feed as a favorite**

Top-level messages in feeds can be marked as favorites on a per user basis only.

You can also search for messages marked as favorites, and remove the favorite designation as needed.

1. Navigate to **Collaborate > Live Feed**.
2. Click either **Group Feeds** or **Record Feeds**, and then click **All Feeds**.

3. In the list of feeds, click the star icon (⭐) adjacent to the feed name you want to mark as a favorite.

   To unfavorite a message, click the star icon (⭐) of a message that is already a favorite.

---

**View a feed**

1. Navigate to **Collaborate > Live Feed**.
2. Click either **Group Feeds** or **Record Feeds**.
   This displays a list of feeds you belong to and an option to view all feeds.
   - To view a feed you belong to, select the feed name.
   - To view a public feed you do not belong to, select **Show All** and then click the feed name. You must be a member to view the feed for a private feed.

---

**Note:** If you navigate away from the live feed application and then back again, the system displays the feed you last visited if you are a member of it.

---

**View a feed member**

1. Navigate to **Collaborate > Live Feed**.
2. Click either **Group Feeds** or **Record Feeds**.
   This displays a list of feeds you belong to and an option to view all feeds.
3. Click the feed name.
   The list of feed members appears, organized by feed administrators and members.

---

**Invite a new member to a group feed**

The feed administrator can invite another user or a group to join a feed.

When a team is added to a feed, all members of the team also indirectly become members of the feed. Only users, and not teams, can be the feed administrators. A team member who is following a feed can become administrator by directly inviting another user to the feed.

1. Navigate to **Collaborate > Live Feed**.
2. Click either **Group Feeds** or **Record Feeds**, and then select the feed name.
3. Click the **Pending members** tab.
4. In the **Add user or team** field, begin entering a user or team name and select a user or team from the suggestion list.
   The invited user or team members receive an email notification.

---

**Invite a new member to a record feed**

The feed administrator can invite another user or a team to join a record feed.

When a team is added, each of the individual members of the team become members of the feed. That is, if Team ABC is added to a feed, the feed does not show that Team ABC is a member. Instead, the feed shows each of the individual members of Team ABC as members of the feed.
When users are invited, their membership state is initially Invited. When they accept the invitation, the system checks their access to the work_notes journal field. If they have write access, they become administrator members; otherwise, they become active members.

1. Navigate to Collaborate > Live Feed.
2. Click Record Feeds and select the record feed name.
3. Click the Pending members tab.
4. In the Add user or team field, begin entering a user or team name and select a user or team from the suggestion list.
   - If you invite a team, the individual members of the team are invited.
   - The invited users receive an email notification.

Remove a member from a feed

The feed administrator can remove any member from the feed.

1. Navigate to Collaborate > Live Feed.
2. Click either Group Feeds or Record Feeds, and then select the feed name.
3. Click the Members tab.
4. Locate the member you want to remove and click Remove.
Note: You can remove members regardless of their current state. That is, you can remove members who have already joined the feed or whose membership is pending.

Subscribe to a feed

Direct members of a feed can subscribe and unsubscribe to the feed in order to receive email notifications whenever any activity in the feed occurs, such as new messages being posted, replies, likes, polls, and @mentions.

Individual indirect members—that is, members of a team that is participating in a feed—cannot subscribe or unsubscribe from a feed directly. If the team administrator subscribes the team to a feed, the individual members receive an email notification for activities in the feed such as new messages being posted, replies, likes, polls and @mentions.

All notifications related to a top-level message, such as replies, mentions, and likes, are sent out with the same subject. The subject line defaults to Live Feed followed by the beginning of the subject line from the top-level message. For example, if the subject of the feed is IT Support, the email notification’s subject will be Live Feed IT Support.

1. Navigate to Collaborate > Live Feed.
2. Click either Group Feeds or Record Feeds, and then select the feed name.
3. Click the subscribe icon (✉).

You can click the unsubscribe icon (✉) to stop receiving email notifications.

Note: If a team is subscribed to receive email notifications for a feed, but a member of the team did not subscribe to the team, that member does not receive the email notifications.

Navigate between live feed and other modules

If you are participating in a feed and navigate to another module, when you return to live feed it displays the feed you were previously viewing if it is public or if it is a private feed that you are a member of.

For example, assume you are viewing a public feed and then navigate to a particular incident record. Upon returning to live feed, the feed you were previously viewing opens. If you were last viewing a private feed that you are not a member of, returning to live feed opens the Company Feed.

Leave a feed

1. Navigate to Collaborate > Live Feed.
2. Click either Group Feeds or Record Feeds, and then select the feed name.
3. Click Leave.

Note: If you leave a feed associated with a record, such as an incident, you automatically rejoin that feed if you modify the associated record again.

Delete a feed

If you are a feed administrator, you can delete the feed.

1. Navigate to Collaborate > Live Feed.
2. Click either Group Feeds or Record Feeds, and then select the feed name.
3. Click **Delete**.
4. Click **Yes** to confirm deletion.

   The feed is deleted from live feed. Messages posted to a private feed are deleted. Messages posted to a public feed remain on the Company Feed without a feed link.

**Live Feed searches**

Two types of searches can be performed in live feed.

Two types of searches can be performed in live feed:

- Simple keyword searches.
- Advanced searches that return content that is relevant to the action you are performing in live feed.

---

**Note:** Many of these features are part of live feed v2, which is available starting with the Fuji release.

---

**Perform a simple search in Live Feed**

All pages have simple keyword search where the keywords apply to the appropriate fields of the entity.

For example:

- When you do a keyword search on feeds, the search is performed on the **Message** field.
- When you search feeds or teams, the search is performed on the **Name** and **Description** fields.
- When you search users you are following or your followers, the search is performed on the **User** field.
- When you are searching for hashtags, the search is performed on the **Hashtag name** field.

To perform a keyword search:

1. Type a word in the **Search messages** field.
2. Click the search icon ( ) to initiate the search.

   All entities that contain the keyword are found.

   Notice that the search results are highlighted below.
Perform an advanced search in Live Feed

If you are working in a feed, you can perform a more advanced search.

To perform an advanced search:

1. Click the down-arrow icon in the **Search messages** field. This displays various filter conditions that are applicable to the entity being searched.
2. Fill in any combination of the fields displayed, and click **Search**.
Note: When you are viewing another user's profile, advanced search is not available.

Notice that the search results are highlighted below.

Remove a Live Feed search filter

When searches are performed, many pages display results that are filtered to provide the most logical information. For example, if you search feeds, the results are filtered to include only feeds of which you are a member. When this occurs, a Filtering is on message is shown.
To remove the filter so you can see the results for all feeds you have access to:

Click the X in the filtering message.

Live Feed UI overview

The Live Feed user interface provides many methods you can use to share content with others in your organization.

Update your Live Feed profile

Your profile information includes a photo or image, and tabs for different types of information including messages, feeds, teams, followers, and those who are following you. You can update your profile information and picture at any time.

**Note:** The functionality described here applies to HTML5-compliant browsers, such as Chrome, Firefox, Safari and IE10 and above.

1. Navigate to **Self-Service > Live Feed**.
2. Click your name or title displayed below your picture.

The profile record opens.
3. Click the pencil icon next to About Me and type a short description about yourself that you want to share with others who view your profile. This is text that you might not necessarily want to change on a regular basis.

4. You can change your profile photo using either of the following two methods:
   - Locate the photo file you want to use, and drag-and-drop it over the existing photo.
   - Hover over the existing picture (or tap the photo in the case of the smartphone or tablet interface) to display the **Upload a picture** link. Click the link, navigate to the location of the photo you want to use, and click **Open**.
Note: Your profile picture is used by the Connect and legacy chat features as well if they are activated.

Work with Live Feed messages

This portion of the UI gives the user access to various types of messages and tools for grouping and categorizing messages. Also included are methods for identifying which user feeds you are following and which users are following you.
### Table 573: Tools for working with live feed messages

<table>
<thead>
<tr>
<th>Menu section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>My Live Feeds</td>
<td>• My Live Feeds: These are custom feeds that allow you to customize live feed to view the content that is most important to you.</td>
</tr>
<tr>
<td>Company Feed</td>
<td>• Company Feed: This feed displays all posts, except those posted to private or unlisted groups.</td>
</tr>
<tr>
<td>Group Feeds</td>
<td>• Group Feeds: These feeds are created by users. Record Feeds: These feeds are associated with a record, such as an incident or change.</td>
</tr>
<tr>
<td>Record Feeds</td>
<td>• Record feeds: These feeds help users collaborate on records by providing a place for anyone who can access the record to share messages and post attachments specific to that record.</td>
</tr>
<tr>
<td>People</td>
<td>• People: This option displays people who are following the current user and people the current user is following.</td>
</tr>
<tr>
<td>Teams</td>
<td>• Teams: This option is used to combine users into groups for the purpose of subscribing to specifically-focused conversations.</td>
</tr>
<tr>
<td>Hashtags</td>
<td>• Hashtags: Hashtags are words marked with a hash (#) symbol in messages. Hashtags are used to categorize messages by keyword or topic for improved search results messages filtered from all conversations that include specific hashtags.</td>
</tr>
</tbody>
</table>

**View another Live Feed user**

When a requester user, one that has no ServiceNow role, is viewing a feed and points to another user's picture, information for the user appears from the Business Card view of the User [sys_user] table.

You can customize ACLs to increase the amount of information you want displayed for users with different roles.
If you click a user's name or title, the user's profile appears. Click the tabs to view different types of information for the user.
Follow other Live Feed users

As you use Live Feed, you can follow the message threads of other users.

1. Navigate to a thread that the user you want to follow is engaged in.
2. Click that user's name or point to the user's picture. A Follow link appears in the user's profile.
3. Click **Follow**.
4. To stop following the user, navigate to **People > Following**.
5. Click **Unfollow**.
6. To view people who are following you, navigate to **People > Followers**.

**Select feeds**

A feed is a stream of related messages. You can use the feed selector to switch between feed types and to view specific group feeds and record feeds.

1. Use the **Search** field to locate any type of feed. As you type in the **Search** field, the results are dynamically displayed. For example, if you type INC, records that are incidents are forced to the top of the results list.
You can also select the following types of feed information.

**Table 574: Feeds types**

<table>
<thead>
<tr>
<th>Feed</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>My Live Feeds</td>
<td>Displays the following posts:</td>
</tr>
<tr>
<td></td>
<td>• messages from users you follow</td>
</tr>
<tr>
<td></td>
<td>• messages with hashtags you follow</td>
</tr>
<tr>
<td>Company Feed</td>
<td>Displays all posts, except those posted to private or unlisted groups.</td>
</tr>
<tr>
<td>Add Group Feed</td>
<td>Allows you to add a group feed.</td>
</tr>
</tbody>
</table>
Follow feeds

You can view users who are following you or other users you are following.

To view users who are following you or other users you are following, click **People** in the sidebar.

The system shows 10 most recent followers or users you are following.

If either list contains more than 10 users, a Show More link appears.

**Note:** If you are following a particular user whose profile changes, that user moves to the front of the Following list. This alerts you to changes in the profile of a user you are following.

Bookmark feeds and hashtags

The live feed interface uses the standard bookmarking functionality for feeds and hashtags.

In UI15 and UI11, you can drag-and-drop group feeds and hashtags from the Feed Summary section to the Edge for quick access.
Add Live Feed to your homepage

You can add live feed to your own homepage or to a global homepage.

1. Navigate to a homepage.
2. Click the add content icon in the top left corner of the homepage.
3. Select Live Feed in the left panel.
4. On the bottom of the window, click Add here in the appropriate layout position, then close the window.
Geneva  ServiceNow  ServiceNow Platform

Note: Administrators can add live feed to a global homepage to make it available for all homepage users by default. Users with any role can add live feed to their homepage; however, administrators can restrict this ability.

Use Live Feed with UI11

Using Live Feed with UI11 requires the UI11-Navpage layout with multiple panes plugin.

1. If you do not see a gray toolbar on the left side of the screen (called your edge), click the Switch to the new UI link in the banner.
2. Navigate to Collaborate > Live Feed.
3. Drag the Live Feed module to the edge.
4. Point to the bookmark on your edge. When the tip window appears, click Edit Bookmark.
5. Select the Flyout check box.
6. Click Update.

- You can now access live feed in a flyout window without navigating away from your working panes.
- Notifications appear on the edge when new messages are posted in your My Feed.

- When you open the bookmark, a blue circle appears beside new messages.
Use teams in Live Feed

Users can be combined into teams for the purpose of subscribing to specifically-focused feeds.

Teams can be created based on any criteria that makes sense for your organization. For example, you can create a team called IT Developers and subscribe the team to feeds such as Coding Standards, AngularJS Development, and ServiceNow Customization Best Practices. When a new developer is added to the IT Developers team, that user automatically has rights to participate in all of the feeds followed by the team.

*Note:* Many of these features are part of live feed v2.

Create a team

When you create a team, you become the team administrator.

As team administrator, you can:

- Modify team properties
- Accept or reject membership requests
- Remove members from the team
- Delete the team

1. Navigate to **Collaborate > Live Feed**.
2. In the sidebar, click your profile picture, then select the **Teams** tab.

3. Click **Create Team**.

4. Enter a **Name** to appear at the top of the feed and in messages posted to the team. This name links to the team feed.

5. Enter a **Description** to appear at the top of the feed and under the team name in the list of all teams.

6. Select the access level for the team:
• **Public Team**: In the list of all teams, any user can see the name of this team. Any user can also see the feed and join the team. Messages in a public team appear in the Company Feed with a link to the team feed.

• **Private Team**: In the list of all teams, any user can see the name of this team, but not the feed. Messages in private teams do not appear on the Company Feed.

When **Private Team** is selected, the **Show this team for anyone to find and request to join** check box is displayed. Select this check box to allow any user to search for this team and request to join it. Clear the check box to allow only invited users to see the feed and join the team.

7. Click **Create**.

The team is created with you as the team administrator.

---

**Note**: A team administrator can create one or more group feeds specifically for the members of this team by clicking the **Group Feeds** tab and **Create Group Feed**. Additionally, the team can be invited to join a feed. Having multiple feeds for the team facilitates focused discussions on topics of interest to the team and allows the team to selectively invite other teams and members to collaborate on specific feeds.

---

**Automatic team creation**

If you create a record on a table that has live feed enabled, such as the Incident table, and click Follow/Show Live Feed for the record, a record feed is created for the record.

The record feed name is based on the table name.

**Join a team**

As a team member, you can:

- View messages posted to the team (team feed). Non-members can also view the feed for a public team.
• View a list of team members.
• Invite another team to join the team.
• Subscribe to team email notifications.
• Leave the team.

1. Navigate to Collaborate > Live Feed.
2. Click Teams and click All Teams.
   • If you received an invitation, click Accept. You can Decline the invitation if you do not want to join the team.
   • If the team is public, click Join.
   • If the team is private, click Request. The team administrator receives an email notification and must accept your request before you can join the team.

View a team feed

1. Navigate to Collaborate > Live Feed.
2. In the sidebar, click your profile picture, then select the Teams tab.
   • To view a team you belong to, select the team name.
   • To view a public team you do not belong to, select All Teams and then click the team name.
     You must be a member to view the feed for a private team.

View a team member

1. Navigate to Collaborate > Live Feed.
2. In the sidebar, click your profile picture, then select the Teams tab.
3. From the team list, select the team name.
   You must be a member of the team to view the members.
4. Click Members.
   The list of team members appears, organized by team administrators and team members.

Invite a new member

You can invite another user to join a team.

1. Navigate to Collaborate > Live Feed.
2. In the sidebar, click your profile picture, then select the Teams tab.
3. From the team list, select the team name.
4. Click the Pending members tab.
5. In the Add User field, enter the name of the user you want to invite. The invited member’s picture appears.
Remove a member

If you are team administrator, you can remove any member from the team.

1. Navigate to **Collaborate > Live Feed**.
2. In the sidebar, click your profile picture, then select the **Teams** tab.
3. From the team list, select the team name.
4. Click the **Members** tab.
5. Click **Remove** below the member name.
6. Click **Remove** again in the confirmation box that opens.

**Note:** You can remove members regardless of their current state. That is, you can remove members who have already joined the team or whose membership is pending.

**Subscribe to team and email notification**

Users can subscribe and unsubscribe to receive email notifications generated by feeds the team participates in.

Email notifications are generated whenever actions such as posting messages, replies, likes, polls, and @mentions are performed.

All notifications related to a top-level message, such as replies, mentions, and likes, are sent out with the same subject. The subject line defaults to **Live Feed** followed by the beginning of the subject line from the top-level message. For example, if the subject of the feed is **IT Support**, the email notification's subject will be **Live Feed IT Support**.

1. Navigate to **Collaborate > Live Feed**.
2. In the sidebar, click your profile picture, then select the **Teams** tab.
3. From the team list, select the team name.
4. Click the subscribe icon (ıc).

© 2017 ServiceNow. All rights reserved. 2420
You can click the unsubscribe icon (:Event:EmailAddress) to stop receiving email notifications.

Subscribe a team to a feed notification

A team administrator can subscribe a team to specific feeds so the members of the team receive email notifications.

1. Navigate to Collaborate > Live Feed.
2. Click Teams and select the team name. You can use the advanced search to locate team of which you are the administrator.
3. Click Feeds.
4. Click the subscribe icon (:Event:EmailAddress) beside the feed name.

You can click the unsubscribe icon (:Event:EmailAddress) to stop receiving email notifications.

Reply to email notifications using email

After receiving an email about a posted message or a reply, you can reply to live feed with your email client.

The body of the reply email becomes a reply to the live feed message. Any attachments included in your reply email are included in the live feed reply.

Leave a team

You can give up membership to a team by leaving that team.

1. Navigate to Collaborate > Live Feed.
2. In the sidebar, click your profile picture, then select the Teams tab.
3. Click Leave beside the team name.

Modify a team

If you are team administrator, you can modify an existing team.

1. Navigate to Collaborate > Live Feed.
2. In the sidebar, click your profile picture, then select the Teams tab.
3. From the team list, select the team name.
4. Click the edit icon (:Event:EmailAddress).
5. Modify the team properties as needed.
6. Click Save.

Manage a membership request

When a user requests membership to a private team, the team administrator receives an email.

If you are the team administrator, you can manage membership requests.

1. Navigate to Collaborate > Live Feed.
2. In the sidebar, click your profile picture, then select the Teams tab.
3. From the team list, select the team name.
If one or more users have been invited to the team, the **Pending members** tab shows the number of invitations sent.

4. Click **Pending members**.
   
   This shows the users who have received invitations, but who have not yet accepted.
5. You can cancel a user’s invitation by clicking **Remove**.
6. You can invite another user to the team by entering their name in the **Add user** field and clicking the Invite member icon (/actions addUser).  

**Delete a team**

If you are the team administrator, you can delete a team.

1. Navigate to **Collaborate > Live Feed**.
2. In the sidebar, click your profile picture, then select the **Teams** tab.
3. From the team list, select the team name.
4. Click the edit icon (actions edit).
5. Click **Delete**.
6. Click **Yes** to confirm deletion.
The team is deleted from live feed. Messages posted to a private team are deleted; messages posted to a public team remain on the company feed without a team link.

If you delete a team that is associated with an active record, such as an incident that has not yet been closed, the team becomes active again when any user modifies that record or follows the record on live feed. The first user who modifies the record becomes the team administrator. The previous messages that existed as part of the record feed are not deleted.

Post content in Live Feed

In Live Feed, you can post new messages and replies to existing messages for all users in the feed. You can also send a reply message to a team or record.

Post a message

You can post a message to open a new feed.

1. Navigate to Collaborate > Live Feed.
2. Click the feed to which you want to post your message, for example My Live Feed, Company Feed, and so on.
3. Enter a message in the Share your thoughts field. Use hashtags, if you want.
4. As needed, click , , or to add a poll, attachment, or screenshot image, respectively.
5. Click Post.

Reply to a live feed message

To participate in an existing thread of a feed, you can compose and submit a reply message. By default, the message is visible to all members of the feed.

1. Navigate to Collaborate > Live Feed.
2. Locate the message you want to reply to.
3. Type your reply in the message box and click Reply.
   As needed, you can add hashtags, attachments, links, or knowledge base articles.
4. To limit who can view the reply to a single individual, select a user from the To choice list.
5. Click Reply.
6. By default, 7 replies are visible for each message. To view additional replies, click Show all replies. Any reply to a reply includes a Threads link.
7. Click Threads to view all of the replies within the selected reply in reverse chronological order within the Message Thread pop-up window, similar to an email thread.
Geneva  ServiceNow  ServiceNow Platform

Message Thread

Navakanth Reddy #suggestion(s) A common black vertical line just before the bars would help with this. Also, line spacing between two options should be more than single-spaced.

Also, the bar says 5 people voted and it shows up 5 people and 1 more. Is this a #bug?

8 days ago • in reply to Nirupam Biswas • Like (1) • Reply • Share • Tags (2)

Nirupam Biswas I guess we need some visual distinction between the poll question and the options. After voting, I got confused for some moments why there are four options in result. See screenshot.

8 days ago • Like (1) • Reply • Share • Tags

Shouvik Goswami We have enabled polls in Live Feed. Please review and provide #suggestions and #feedbacks

How do you like the poll in Live Feed?

- Exactly what I was looking for
- Wanted something more
- Did not visualize it this way

Vote

8 days ago • Like (1) • Reply • Share • Tags (2)
Attach a file to a post or reply

There are two methods that can be used to attach files to posts, drag-and-drop and using the paperclip icon ( ). Additionally, you can paste images into a post or reply from the clipboard.

1. Do one of the following:
   - While composing a post or reply (see Post a live feed message or Reply to a live feed message), locate one or more images or other supported types of files, and drag them into the message box.
   - Click the paperclip icon ( ) in the message box, navigate to the location of the files you want to attach, select them, and click Open.

2. To add more files, drag-and-drop additional files into the text box below the displayed files or images, or click the plus sign.
3. If you want to delete an attachment before posting the message, click the trashcan icon adjacent to the attachment.

4. When you have finished entering your message, click Post or Reply.

5. If you added files that do not contain previews and then posted the message, they will be listed in the message box with a Download link.

6. Click the link to download the files.
Copy an image from the clipboard

In addition to adding files to a message, you can also copy images to the clipboard and paste them into a message or reply.

1. Click into the Comment box and add a comment.

2. Click the Add an image button. The Screenshots dialog box opens.

3. Right-click the Paste an image here box and select Paste to paste the image from the clipboard.
4. To add another image, repeat the previous steps.
5. When you finish pasting images into the message, click Done.
6. To change the link text, click Pasted Image and enter a new label.
7. When you finish, click Post or Reply.
8. To remove an image, click the trash can icon adjacent to the link.
9. To make changes to an existing image, click the edit icon (📝).

Note: The copy/paste functionality is supported in Chrome, Firefox, and Internet Explorer 11.
Add a link to a post or reply

1. Click into the Comment box. The Add a link button appears.
2. Click the Add a link button. The URL and Link text fields appear.

![Comment box with Add a link button](image)

3. Enter the URL and the text you want to appear as the link.
4. To add another link, click the link icon under the URL field.
5. When you have finished entering your message, click Post or Reply.

![Comment box with links](image)

---

**Note:** To remove a link, click the trash can icon adjacent to the link. To make changes to an existing link, click the edit icon ( ).
Add a knowledge article link to a post or reply

1. While composing a post or reply, type the ID of the knowledge article you want to link to. It is not necessary to identify the knowledge article ID as a hyperlink; live feed recognizes it as such and automatically converts it into a link when the message is posted.
2. When you are finished entering your message, click Post or Reply.

Add a poll to a message

When you are posting a message, you have the option of creating a poll in order to record your viewers' opinions. As users take the poll, their feedback statistics are displayed within the feed.

1. Open the feed where you want to create a message with a poll.
2. Enter the message with an invitation to take the poll in the Share your thoughts box.
3. Click the poll icon ( ).
4. In the **Question** field, enter the question on which you are soliciting opinions.  
5. In the **Option** fields, enter possible responses to the query.  
6. To add more response options, click **Add more options**.  
7. Click **Post**.  
   After the poll is posted, users who view the message can vote. After they vote, their pictures appear next to their response, with others who have voted for the same option.

**Note:** Only the poll creator can view results without casting a vote. Other members can see results only after casting a vote or if the poll is closed.
An email notification with the poll result is sent to all participants of the poll informing them that the poll is closed.

Add mentions to a message

An @mention is any posted update that contains @username anywhere in the body of the message. The user’s name, with a link to that user’s feed, is inserted into the message. Additionally, the mentioned user receives an email notification about the mention. This is a great way of bringing attention to other members in a conversation. All @mentions are included in the logged-in user’s My Feed.

**Note:** If an @mention is used by a member of a team, only members of that team appear in the auto-suggest list.

1. While entering a message or reply in a conversation, type @username anywhere in the message. As you type, an auto-suggest list appears with names and pictures of users that match your entries. For example, if you type @t, the auto-suggest list shows the pictures and names of all users with names that start with T.
2. Click the user you want to add.
That user’s name is inserted into the @mention in the body of the message.

Note: My Feed displays @mentions of you and the teams you are a member of. Profile feeds for another user display only @mentions of that user.

Like a message

1. Navigate to Collaborate > Live Feed.
2. Click Feeds.
3. Locate the feed that contains the message you want to like.
4. Under the message title, click Like.
   If the message has previously been liked, the number of likes for a message displays next to the Like link.
   For a top-level message, the profile pictures of the first 5 users who liked a message are displayed.
   Any additional users who liked the message can be viewed by hovering over the # more link. If the logged-in user has liked the message, that user’s image will appear first, followed by the image of the user who last liked the message.
   For a reply, you must hover your mouse pointer over the count link to view the users who liked the message.

   Note: After a message has been liked, it cannot be unliked.

Delete a message

Users can delete any of their own posts or replies. A conversation administrator can also delete messages.
1. Hover over the message you want to delete.
2. Click the trash can icon in the upper-right-hand corner of the message box.
3. Click Delete in the confirmation box.
   The message disappears from the feed.

Platform security

Security is built into all levels of functionality. Implement the security features that are most appropriate for your organization, from managing failed logins and encrypted password protection, to roles and access control rules, to audit logs of user interactions.

General platform security

General security refers to the properties, settings, and options that you can use to better secure your instance.

General security settings properties

Security settings provides several properties to control the level of security on your instance.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Escaping and embedded script support</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>glide.ui.security.allow_codetag</td>
<td>Allow support for embedding HTML code by using the [code] tag.</td>
<td>True</td>
</tr>
<tr>
<td>glide.ui.security.codetag.allow_script</td>
<td>Allow embedded HTML (using [code] tags) to contain Javascript tags</td>
<td></td>
</tr>
<tr>
<td>glide.ui.escape_all_script</td>
<td>Forces all expressions within Jelly <code>&lt;script&gt;</code> tags to be escaped by default.</td>
<td></td>
</tr>
<tr>
<td><strong>Attachment Limits and Behavior</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.glide.attachment.max_size</td>
<td>Maximum file attachment size in megabytes:</td>
<td></td>
</tr>
<tr>
<td>glide.attachment.role</td>
<td>List of roles (comma-separated) that can create attachments:</td>
<td></td>
</tr>
<tr>
<td>glide.attachment.extensions</td>
<td>List of file extensions (comma-separated) that can be attached to documents via the attachment dialog. Extensions should not include the dot (.) e.g. xls,xlsx,doc,docx. Leave blank to allow all extensions.</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>glide.ui.strict_customer_uploaded_static_content_types</td>
<td>When set to 'true' turns on the ability to restrict the types of files that can be downloaded, when they have been uploaded using the Upload File functionality of the platform. Used in conjunction with glide.ui.strict_customer_uploaded_content_types.</td>
<td></td>
</tr>
<tr>
<td>glide.ui.strict_customer_uploaded_content_types</td>
<td>glide.ui.strict_customer_uploaded_static_content_types parameter includes a list of comma delimited files. These will be the only file types that can be downloaded as static content from an instance.</td>
<td></td>
</tr>
<tr>
<td>glide.ui.attachment.force_download_all_mime_types</td>
<td>Forcing download of all attachment files.</td>
<td></td>
</tr>
<tr>
<td>glide.security.file.mime_type.validation</td>
<td>This property must be set to activate MIME type checking for uploads (All version Eureka and up). Enables (true) or disables (false) mime type validation for file attachments. File extensions configured via glide.attachment.extensions will be checked for MIME type during upload.</td>
<td></td>
</tr>
</tbody>
</table>

**Security Manager and Options**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.security.manager</td>
<td>Security Manager</td>
</tr>
<tr>
<td>glide.sm.default_mode</td>
<td>Security manager default behavior in the absence of any ACLs on a table</td>
</tr>
<tr>
<td>glide.security.strict.updates</td>
<td>Double check security on inbound transactions during form submission (rights are always checked on form generation)</td>
</tr>
<tr>
<td>glide.security.strict.actions</td>
<td>Check conditions on UI actions before execution, normally the conditions are only checked during form rendering</td>
</tr>
<tr>
<td>glide.security.granular.create</td>
<td>Enforce create (as opposed to write) rules on new records.</td>
</tr>
<tr>
<td>glide.security.explain.write.locks</td>
<td>Display an explanation on locked form elements.</td>
</tr>
</tbody>
</table>

**Cookies**
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.forgetme</td>
<td>Remove &quot;Remember me&quot; checkbox from login page.</td>
<td></td>
</tr>
<tr>
<td>glide.ui.secure_cookies</td>
<td>Enable secure session cookies: Enable additional cookie security. If checked, strict session cookie validation is enforced. With version 3 cookies enabled, additional security requirements are also enforced.</td>
<td></td>
</tr>
<tr>
<td>glide.secure_cookie.debug</td>
<td><strong>Secure session cookie debugging:</strong> Check to enable extensive debug logging of secure session cookie operations.</td>
<td></td>
</tr>
<tr>
<td>glide.script.use.sandbox</td>
<td>Run client generated scripts (AJAXEvaluate and query conditions) inside of a reduced rights &quot;sandbox&quot;. If enabled, only those business rules and script includes with the &quot;Client callable&quot; checkbox set to true are available and certain backend API calls are disallowed.</td>
<td></td>
</tr>
<tr>
<td>glide.script.allow.ajaxevaluate</td>
<td>Enable the AJAXEvaluate processor</td>
<td></td>
</tr>
<tr>
<td>glide.script.secure.ajaxgliderecord</td>
<td>Apply standard security ACLs to AJAXGlideRecord calls</td>
<td></td>
</tr>
<tr>
<td>com.glide.communications.trustmanager_trust_all</td>
<td>By default, the instance trusts a certificate's Certificate Authority (CA). This ensures the instance accepts self-issued certificates. If you want to validate a certificate's CA, set the system property to false.</td>
<td></td>
</tr>
<tr>
<td>glide.outbound.sslv3.disabled</td>
<td>When active, outbound connections from an instance will be forced to use TLS instead of SSL.</td>
<td></td>
</tr>
</tbody>
</table>

**High Security Settings**

High Security Settings provide advanced security options for your instance.
These features are available:

- **Default property values**: to harden security on your platform by centralizing all critical security settings to one location for management and auditing.
- **Default deny property**: provides a security manager property to control the default security behavior for table access.
- **Security Administrator role**: provides a role to prevent modification of key security settings and resources. The Security Administrator role is not inherited by the admin role and must be explicitly assigned.
- **Elevated privilege**: allows users with the security admin role to operate in the context of a normal user and elevate to higher security role when needed.
- **Property access control**: allows security administrators to set the roles required to read and write properties.
- **Transaction and system logs**: are read only.
- **Access control rules**: control what data users can access and how they can access it.

High Security Settings automatically activates the Contextual Security plugin if it is not already active. In addition, Platform Security Settings - High delivers the following settings and features in the context of increasing the security of your instance.

### Property access control

Two additional columns are created in the Properties [sys_properties] table.

- **read_roles**: a comma-separated list of role names that are allowed to read all fields of this property
- **write_roles**: a comma-separated list of role names that are allowed to write/modify all fields of this property

Properties listed in the Properties table have **read_roles** of admin, and **write_roles** of security_admin. This means that users with the admin role can view and read the property values, but must elevate to the security_admin role to modify them.

### Notifications

Activation of high security settings also activates security warning messages. The following is an example of a message that appears after an approval.
Figure 679: Security Warning notification

High Security Settings properties

High Security Settings provides several properties to control the level of security on your instance.

Table 576: High Security Settings properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.escape_text</td>
<td>Escape XML values at the parser level for the user interface. This will prevent reflected and stored cross site scripting attacks. Default: Yes</td>
</tr>
<tr>
<td>glide.ui.escape_all_script</td>
<td>Forces all expressions within Jelly <code>&lt;script&gt;</code> tags to be escaped by default. Default: No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.ui.rotate_sessions</td>
<td>Rotate HTTP session identifiers to reduce security vulnerabilities. See: <a href="http://www.owasp.org/index.php/Session_Management#Rotate_Session_Identifiers">http://www.owasp.org/index.php/Session_Management#Rotate_Session_Identifiers</a>.</td>
</tr>
<tr>
<td></td>
<td>If you are using the SAML 2.0 plugin for Single Sign-on authentication, set this feature to false. Otherwise, it interferes with the session information sharing that takes place between the instance and the Identity Provider.</td>
</tr>
<tr>
<td>glide.ui.secure_cookies</td>
<td>Enable secure session cookies: Enable additional cookie security. If selected, strict session cookie validation is enforced.</td>
</tr>
<tr>
<td>glide.security.strict.updates</td>
<td>Double check security on inbound transactions during form submission (rights are always checked on form generation).</td>
</tr>
<tr>
<td>glide.security.strict.actions</td>
<td>Check conditions on UI actions before execution; normally the conditions are only checked during form rendering.</td>
</tr>
<tr>
<td>glide.security.use_csrf_token</td>
<td>Enable usage of a secure token to identify and validate incoming requests. This token is used to prevent cross site request forgery attacks.</td>
</tr>
<tr>
<td>glide.ui.escape_html_list_field</td>
<td>Escape HTML for HTML fields in a list view.</td>
</tr>
<tr>
<td>glide.html.escape_script</td>
<td>Escape JavaScript tags in HTML fields.</td>
</tr>
<tr>
<td>glide.ui.forgetme</td>
<td>Remove Remember me check box from login page.</td>
</tr>
<tr>
<td>glide.smtp.auth</td>
<td>Authenticate with the SMTP server by the user name and password properties.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.script.use.sandbox</td>
<td>Run client generated scripts (AJAXEvaluate and query conditions) inside of a reduced rights &quot;sandbox&quot;. If enabled, only those business rules and script includes with the Client callable checkbox set to true are available and certain back-end API calls are disallowed.</td>
</tr>
<tr>
<td>glide.soap.strict_security</td>
<td>Enforce strict security on incoming SOAP requests. Checking this requires incoming SOAP requests to go through the security manager for table and field access and checks SOAP users for the correct roles for using the web service.</td>
</tr>
<tr>
<td>glide.basicauth.required.wsdl</td>
<td>Require authorization for incoming WSDL requests.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you choose not to require authorization for incoming WSDL requests, you will need to modify the Access Control (ACL) rules to allow guest users to access the WSDL content.</td>
</tr>
<tr>
<td>glide.basicauth.required.csv</td>
<td>Require basic authorization for incoming CSV requests.</td>
</tr>
<tr>
<td>glide.basicauth.required.excel</td>
<td>Require basic authorization for incoming Excel requests.</td>
</tr>
<tr>
<td>glide.basicauth.required.importprocessor</td>
<td>Require basic authorization for incoming import requests.</td>
</tr>
<tr>
<td>glide.basicauth.required.pdf</td>
<td>Require basic authorization for incoming PDF requests.</td>
</tr>
<tr>
<td>glide.basicauth.required.rss</td>
<td>Require basic authorization for incoming RSS requests.</td>
</tr>
<tr>
<td>glide.basicauth.required.scriptedprocessor</td>
<td>Require basic authorization for incoming script requests.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.basicauth.required.soap</td>
<td>Require basic authorization for incoming SOAP requests.</td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong> Yes</td>
</tr>
<tr>
<td>glide.basicauth.required.unl</td>
<td>Require basic authorization for incoming unload requests.</td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong> Yes</td>
</tr>
<tr>
<td>glide.basicauth.required.xml</td>
<td>Require basic authorization for incoming XML requests.</td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong> Yes</td>
</tr>
<tr>
<td>glide.basicauth.required.xsd</td>
<td>Require basic authorization for incoming XSD requests.</td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong> Yes</td>
</tr>
<tr>
<td>glide.cms.catalog_uri_relative</td>
<td>Enforce relative links from the URI parameter on /ess/catalog.do. If checked, then only relative URLs are permitted through the /ess/catalog.do page using the parameter 'uri'. If unchecked, all URLs are permitted, which may permit linking to external unauthorized content.</td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong> Yes</td>
</tr>
<tr>
<td>glide.set_x_frame_options</td>
<td>Enable this property to set the X-Frame-Options response header to SAMEORIGIN for all UI pages. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a &lt;frame&gt; or &lt;iframe&gt;. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites. <a href="https://developer.mozilla.org/en/the_x-frame-options_response_header">https://developer.mozilla.org/en/the_x-frame-options_response_header</a></td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong> Yes</td>
</tr>
<tr>
<td>glide.ui.attachment.download_mime_types</td>
<td>A list of comma separated attachment mime types that do not render inline in the browser. This will prevent cross site scripting attacks. For example, text/html forces HTML files to be downloaded to the client as attachments rather than viewed inline in the browser.</td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong></td>
</tr>
<tr>
<td>glide.security.groupby_acl_check</td>
<td>When this property is enabled, for GroupBy operations ACL checks are performed for the &quot;group&quot; names based on the actual data from the groups.</td>
</tr>
<tr>
<td></td>
<td><strong>Default:</strong> Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>glide.security.diag_txns_acl</td>
<td>If it is set to true, only admin user or user from allowed ip address can access stats.do, threads.do and replication.do. Default: Yes</td>
</tr>
<tr>
<td>glide.ui.security.allow_codetag</td>
<td>Allow support for embedding HTML code by using the [code] tag. Default: Yes</td>
</tr>
<tr>
<td>glide.ui.security.codetag.allow_script</td>
<td>Allow embedded HTML (using [code] tags) to contain JavaScript tags. Default: No</td>
</tr>
<tr>
<td>glide.script.allow.ajaxevaluate</td>
<td>Enable the AJAXEvaluate processor. Default: No</td>
</tr>
<tr>
<td>glide.login.autocomplete</td>
<td>Allow browsers to use auto-complete on password fields on login forms. Default: No</td>
</tr>
<tr>
<td>glide.security.csrf_previous.allow</td>
<td>Allow usage of an expired secure token to identify and validate incoming requests. This token is used to prevent cross site request forgery attacks. Default: No</td>
</tr>
<tr>
<td>glide.security.csrf_previous.time_limit</td>
<td>Time in seconds for a secure token to expire. It allows control over the length of time that the previous CSRF token is valid. When the user session expires, the secure token expires with it unless the &quot;allowing reuse of expired tokens are allowed&quot; property is enabled and it's within the time frame described by this property. This token is used to prevent cross-site request forgery attacks. Default: 86400 seconds or 1 day</td>
</tr>
<tr>
<td>glide.security.csrf.strict.validation.mode</td>
<td>This property enforces strict validation on CSRF tokens so that users cannot resubmit a request if the CSRF token does not match. Default: false</td>
</tr>
<tr>
<td>glide.basicauth.required.schema</td>
<td>Require basic authentication for inbound table schema requests. Default: true</td>
</tr>
</tbody>
</table>

These are defined in the sys_properties table, but are not visible on the High Security Settings page.
Default deny property

Activating the High Security plugin creates the glide.sm.default_mode security property, which controls the security manager default behavior when the only matching ACL rules are the wildcard table ACL rules.

The High Security application also includes a set of wildcard table ACL rules for the most common record-based operations: read, write, create, and delete as well as a significant number of ACLs to provide role-based access to system tables. For example, there are ACLs that grant sys_script access to the business_rule_admin role because that role is documented as being able to manage business rules.

The choices for the glide.sm.default_mode property are:

- **Deny Access**: The wildcard table ACL rules restrict the read, write, create, and delete operations on all tables unless the user has the admin role or meets the requirements of another table ACL rule. Other operations, such as report_on and personalize_choices, are unaffected by this setting.

- **Allow Access**: The wildcard table ACL rules allow the read, write, create, and delete operations on all tables unless there are specific table ACL rules in place to restrict such operations.

  **Note**: By default, the wildcard table ACL rules are the only ACL rules that check for the value of the glide.sm.default_mode property. If you want to control other operations with this setting, create your own ACL rules to check for this property value.

Security Administrator role

When the High Security plugin is activated, a new role called security_admin is created and added to the default System Administrator user.

This role is a peer to the admin role and, therefore, is not inherited by users who are assigned the admin role by default (base system System Administrator, for example). This new role is marked as an elevated privilege, which means the user who is assigned the role will need to be manually elevated to the role during an interactive session. The security_admin role protects resources in the platform such as ACL, properties, and records, that require security to bar access by the normal admin user.

Elevated privilege

A role that requires elevated privilege prevents the system from assigning it to a user at login. Instead, a user must manually elevate privileges to receive the elevated role.

In the base system, only the security_admin role requires elevated privilege.
Figure 680: The security administrator role

**Note:** The record for the system_admin role is only visible to users who elevate privileges to the security_admin role.

An elevated privilege role only lasts for the duration of the user session. Session timeout or log-out removes the role. To use an elevated role, all these conditions must be met:

- The role must be assigned to the user.
- The user must manually elevate roles.

User who do not have an elevated privilege role do not see an option to elevate roles. For example in the base system, only the System Administrator user has the security_admin role.
Figure 681: Roles assigned to the System Administrator user

Therefore in the base system, only the System Administrator user has the option to **Elevate Roles**.
**Elevate to a privileged role**

The out-of-box admin can elevate to a privileged role to have access to the features of High Security Settings.

Role required: admin

Note: If you grant additional users the admin role, they cannot elevate to a privileged role. Only the out-of-box admin can elevate.

1. Do the appropriate action for your version of the UI:

<table>
<thead>
<tr>
<th>UI16</th>
<th>1. In the banner frame, click your user name.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. Select <strong>Elevate Roles</strong>.</td>
</tr>
<tr>
<td>UI15 or UI11</td>
<td>In the banner frame, click the lock icon (🔒) by your user name.</td>
</tr>
</tbody>
</table>

A dialog box appears.

![Activate an Elevated Privilege](image)

2. Select an elevated role and click **OK**. This grants the user elevated privileges to all resources controlled by the role for the remainder of the session. When the user logs out, the elevated privileges are terminated and must be reestablished at the next login. When elevated privileges are activated, the icon has an unlocked appearance.
Force administrators to manually elevate
A property is available to force all users with the administrator role to manually select the role that they want to elevate to.

Role required: security_admin

1. Login as a user with the security_admin role.
2. Elevate roles to security_admin.
3. Navigate to sys_properties.list.
4. Find the glide.security.strict_elevate_privilege property.
5. Set the property to true.

Activate High Security Settings
The High Security Settings plugin is active by default on all new instances.

Role required: security_admin

If it is not active on your instance, you can request the plugin.

1. In the HI Service Portal, click Service Catalog > Activate Plugin.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
</tbody>
</table>
Specify the date and time you would like this plugin to be enabled

Date and time must be at least 2 business days from the current time.

Note: Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.

Reason/Comments

Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows.

3. Click Submit.

Certificates

The instance uses certificates to establish secure connections and validate signatures.

Certificates are used for features such as:

- **LDAPS**
- **Mutual authentication**
- **Web services security**
- **MID Server**

In order to use a certificate, you need to generate or purchase a certificate for the secured server or client and upload it to an instance.

**LDAP certificates**

Uploading an SSL certificate allows the instance to establish an LDAP over SSL (LDAPS protocol) connection with an LDAP server.

The instance accepts two types of LDAP certificates:

<table>
<thead>
<tr>
<th>Certificate</th>
<th>Type</th>
<th>Required for</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP server certificate</td>
<td>Any supported type</td>
<td>All LDAP configurations</td>
</tr>
<tr>
<td>LDAP client certificate</td>
<td>Java keystore type</td>
<td>Mutual authentication</td>
</tr>
</tbody>
</table>

If there are multiple server certificates, ServiceNow tries each server certificate in turn until the LDAP server allows the connection. If you use multiple LDAP servers, be sure to include the SSL certificate for each LDAP server.

If your LDAP server requires **mutual authentication** (requires the client to present a certificate in addition to the server), you must also provide your LDAP server’s client certificate in a Java keystore type certificate.
Upload a certificate to an instance

Administrators can add a certificate to the instance from the Certificates module.

Role required: admin

Note: When a certificate is updated on the ADFS server, you also need to upload an updated certificate to the instance.

1. Navigate to System Definition > Certificates.
2. Click New.
3. Fill in the following fields (see table).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify a unique name for the certificate.</td>
</tr>
<tr>
<td>Expiration notification</td>
<td>[Optional] Select whether you want to send a notification when the certificate is about to expire.</td>
</tr>
<tr>
<td>Active</td>
<td>Select whether the instance should use this certificate for secure communications and signing requests.</td>
</tr>
<tr>
<td>Short Description</td>
<td>[Optional] Enter a text description of the certificate such as the requester or server name.</td>
</tr>
<tr>
<td>Format</td>
<td>Select the certificate format. The instance supports the PEM and DER formats. See Certificates.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the certificate container. The instance recognizes certificates from trust stores, Java keystore, and PKCS#12 keystores.</td>
</tr>
<tr>
<td>PEM Certificate</td>
<td>Enter the base-64 encoded PEM-formatted text containing the DER certificate. The instance decodes the certificate to populate the Valid from, Expires, Expires in days, Issuer, and Subject fields.</td>
</tr>
</tbody>
</table>

4. Click Submit.

During the upload, the module extracts and displays the certificate's read-only properties in these fields:
- Valid from date
- Expiration date
- Issuer
- Subject of the certificate

5. Validate the certificate or keystore.
Certificate criteria
Certificates must meet several criteria.

A valid certificate must meet these criteria:

- The certificate can have a key size up to 2048 bits.
- The certificate must have one of these file extensions:

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DER</td>
<td>The Distinguished Encoding Rules format is a binary message transfer syntax. This format also supports the .CER and .CRT file extensions.</td>
</tr>
</tbody>
</table>
### Certificate trust

By default, ServiceNow trusts a certificate's Certificate Authority (CA).

This ensures the instance accepts self-issued certificates. You must set the system property `com.glide.communications.trustmanager_trust_all` to `false`. If you do not set the property false, the instance trusts any certificate.

#### Generate an LDAP client certificate

Generate an LDAP client certificate for mutual authentication using OpenSSL. The final output is a PKCS#12 certificate stored within a Java keystore.

Role required: admin

See the [OpenSSL documentation](https://www.openssl.org) for more information about generating certificates.

These steps assume you have access to OpenSSL.

Enter these commands in a command line interface.

1. **Generate a self-signed client certificate.**

   For example, this command creates a client certificate `test1-cert.crt` based on the `test1-key.key` private key.

   ```bash
   openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout test1-key.key -out test1-cert.crt
   ```

2. **Convert both the certificate file and private key to PKCS#12 (a file with a .pfx or .p12 extension).**

   For example, this command converts the client certificate and private key to a PKCS#12 certificate called `test1-certificate.pfx`.

   ```bash
   openssl pkcs12 -export -out test1-certificate.pfx -inkey test1-key.key -in test1-cert.crt
   ```

3. **Generate the Java Key Store and import the pkcs12 file into it.**

   For example, this command imports the certificate to the `test1.jks` Java keystore.

   ```bash
   keytool -importkeystore -srckeystore test1-certificate.pfx -srcstoretype PKCS12 -destkeystore test1.jks
   ```

4. **Upload the certificate** in the keystore file (`test1.jks`) to the instance.

#### Generate a server certificate

You can use keytool to generate a new Java keystore file, create a certificate signing request (CSR), and import the private key, public certificate pair, and signed certificates into the keystore.

Role required: admin

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CER</td>
<td>A certificate file extensions for certificates using the Distinguished Encoding Rules format.</td>
</tr>
<tr>
<td>CRT</td>
<td>A certificate file extensions for certificates using the Distinguished Encoding Rules format.</td>
</tr>
<tr>
<td>PEM</td>
<td>The <em>Privacy Enhanced Mail</em> format is a base-64 encoded DER certificate enclosed between &quot;-----BEGIN CERTIFICATE-----&quot; and &quot;-----END CERTIFICATE-----&quot; text strings.</td>
</tr>
</tbody>
</table>
See the Java keytool documentation for more information on generating keys and CSRs.

Enter these commands in a command line interface:

1. Generate a Java keystore and key pair.
   For example, this command creates a keystore called my.keystore and generates a private key called mydomain within the keystore.

   ```
   keytool -genkey -alias mydomain -keyalg RSA -keystore my.keystore
   ```

2. Generate a CSR for an existing Java keystore.
   For example, this command generates a CSR called mydomain.csr or the mydomain key.

   ```
   keytool -certreq -alias mydomain -keystore my.keystore -file mydomain.csr
   ```

3. Import a root or intermediate certificate authority, or CA, certificate to the Java keystore.
   For example, this command imports the CA certificate for Thawte. This command assumes that Thawte was the CA that signed the CSR.

   ```
   keytool -import -trustcacerts -alias root -file Thawte.crt -keystore my.keystore
   ```

4. Import a signed primary certificate to the Java keystore.
   For example, this command imports the signed certificate mydomain.crt into the keystore.

   ```
   keytool -import -trustcacerts -alias mydomain -file mydomain.crt -keystore my.keystore
   ```

5. Upload the certificate in the keystore file (my.keystore) to the instance.

   **Upload a trusted server certificate**
   By uploading the service provider’s trusted server certificate, the instance ensures it is connecting to a valid and secure service.

   Role required: admin

   The instance validates outbound Web Service calls by using the certificate provided by the service provider.

   1. Create a new Certificate record with the type “Trust Store Cert”.
   2. Do one of the following actions:
      - Attach the service provider’s DER formatted certificate.
      - Copy and paste the service provider’s PEM format certificate into the **PEM Certificate** field.
Validate a certificate or a keystore
Administrators should validate certificates and keystores after uploading them to determine if there are any issues to resolve.

Role required: admin

If the instance encounters any errors with the certificate or keystore, it displays an error message.

1. Navigate to System Definition > Certificates.
2. Select the certificate or keystore you want to validate.
3. From the X.509 Certificate form, click the Validate Stores/Certificates related link.
   For example, this certificate fails validation because it is expired.
Extended table security

Most ITIL tables are part of a table hierarchy which enables similar tables to share a common set of fields in a parental table.

For example, the Task table is the parent of both Incident and Problem (and many others).

When you are securing a system, it's important to understand how the security model interacts with the table hierarchy. It's also important to know which security manager you are running.
Determine the current security manager

Most ITIL tables are part of a table hierarchy which enables similar tables to share a common set of fields in a parental table.

Role required: security_admin or admin

For example, the task table is the parent of both incident and problem (and many others).

When you're securing a system, it's important to understand how the security model interacts with the table hierarchy. It's also important to know which security manager you are running.

2. Look at the property labeled Security Manager.
   - If it reads com.glide.sys.security.ContextualSecurityManager, you are running the Contextual Security Manager.
   - If it is blank, you are running the Simple Security Manager.

Simple security manager

The simple security manager protects fields via their dictionary entries.

**Note:** Functionality described here requires the Admin role.

To secure a field, bring up that field's dictionary entry and add read/write/delete roles to the field. Row level operations (like insert or delete) are implemented by putting roles on the table's sys_dictionary entry.

Since the simple security manager is based on fields themselves, it has no concept of, or interaction with, the system's table hierarchy. If a field is defined on task, it will be secured identically whether that field is actually part of an incident, a change, or a problem.

Contextual security manager

Unlike the simple security manager, the contextual security manager is aware of the system's table hierarchy, which means you can potentially have different security rules for a field based on where in the hierarchy it is displayed.

**Note:** Functionality described here requires the Admin role.

**Note:** This section assumes you are familiar with the contextual security manager and its ACL base functionality.

The basic rules to remember with the contextual security manager is that it looks up the table hierarchy for rules until it finds one, then stops looking. So in the case of a task field like short_description, the system would look for rules in different ways based on where that field was being used.

When looking at an incident, the system would look for rules in the following order:

incident.short_description, task.short_description, *.short_description

When looking at a change_request, the system would look for rules in the following order:

change_request.short_description, task.short_description, *.short_description
What this means is that a rule at a lower level in the hierarchy can override a rule on a base table. Perhaps I am allowed to change the `short_description` on incidents, but not on change requests. This lends itself to a number of best practices.

**Contextual Security Manager Best Practices**

Secure as high in the hierarchy as you can

It's easier to secure `short_description` on task than it is to secure it on incident, and change_request, and problem, and sc_request, etc. There's only one rule to write for one thing, and if you have to change it, there is only one rule to change. You don't lose any flexibility here, since if you end up needing to secure incident differently, you can still put a specific rule on `incident.short_description` and still have all the other extensions use the rule on `task.short_description`

Use as few rules as possible

This is a corollary to the previous suggestion, but fewer rules are easier to manage. There's no real efficiency difference in terms of processing time between having a rule in incident, another one on problem, and third one on change_request vs. having a single rule on task, but there is a big efficiency different in terms of human time in creating and maintaining the rule base.

But not too few

If you do find yourself needing different security constants for `incident.short_description` vs. `task.short_description`, you could conceivably handle the difference with a scripted rule on `task.short_description` that checked the class name of the current record and responded differently in each case. While such an approach will work, it usually results in fairly large and complicated script rules (especially after they've gone through multiple revisions). You'll usually have an easier-to-manage rule base if you put one simple rule on task and a second simple rule on incident.

**Additional security options**

The application has a wide variety of security options to choose from.

Depending on the security requirements of your particular deployment, it might make sense to run the system with all of its security options enabled. Some of the options make the system more secure, but can offer additional complexity from an implementation standpoint.

All security settings in this page are configured in **System Properties > Security**.

**Remember me login cookie**

When you log on to the instance, you have the option to select the Remember me check box on the login screen, which stores a cookie on your browser.

This cookie contains information that the application server can use to authenticate your user credentials automatically the next time you visit the instance. For most users, this is a highly convenient feature.

In some deployments, however, it is desirable to have users authenticate with the system every time they connect. To support this, the system lets you disable the Remember me functionality by removing the corresponding check box (and its resultant cookie) from the login screen.

The default value for the Remember me check box can also be changed.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove Remember me* checkbox from login page</td>
<td>glide.ui.forgetme</td>
<td>Property disabled (Remember me check box is visible)</td>
</tr>
</tbody>
</table>
You can also modify the session timeout, forcing users to reauthenticate if they do not continue interacting with the instance. Session timeout only works if the Remember me check box in the login screen is not selected.

**Double-check form submission**

When the system determines that a particular field (such as task.number) should not be written to by the current user, the system renders that field in a read-only mode, which is why the number field is not writable on most incidents.

If you set the system to double-check the values of any incoming fields for writability, then the system will apply the same set of security rules to the inbound leg of a transaction. When you submit an incident, for example, the system double-checks to determine if the number field can be written to before posting any changes.

If you tell the system not to double-check inbound transactions, then the system will allow you to write to a nominally read-only field if that is the transaction the client sends back. In many deployments this is actually a desirable behavior if, for example, you are using client scripts to set nominally read-only fields in response to user selections in other, writable fields.

<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double check security on inbound transactions during form submission (rights are always checked on form generation)</td>
<td>Disabled (no double checking)</td>
</tr>
</tbody>
</table>

**Script sandboxing**

There are two cases within the system that allow the client to send scripts to the server for evaluation.

- Filters and/or queries: It is legal to send a filter to the server which reads something like this - assigned_to=javascript:getMyGroups().
- System API: The API call AJAXEvaluate allows the client to run arbitrary scripts on the server and receive a response.

If you enable script sandboxing, the script being evaluated via either of these two entry points runs within a reduced rights sandbox with the following characteristics:

- Only those business rules marked client callable are available within the sandbox.
- Only those script includes marked client callable are available within the sandbox.
- Certain API calls (largely but not entirely limited to those dealing with direct DB access) are not allowed.
- Data cannot be inserted, updated, or deleted from within the sandbox. Any calls to current.update(), for example, are ignored.

If you run the system without script sandboxing enabled, then none of these restrictions apply.

**Note:** This property is activated by default when you activate the High Security Settings plugin. Do not activate this property outside of the plugin.
<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run client generated scripts <em>(AJAXEvaluate and query conditions)</em> inside of a reduced rights &quot;sandbox&quot;. If enabled, only those business rules and script includes with the <strong>Client callable</strong> checkbox set to true are available and certain back-end API calls are disallowed.</td>
<td>Enabled (sandbox in use)</td>
</tr>
</tbody>
</table>

Enable the AJAXEvaluate processor

As was mentioned in the section on script sandboxing, the **AJAXEvaluate** API call allows the client to send, and execute, arbitrary script on the server.

For an additional level of security, over and above that offered by the script sandbox, it is possible to entirely disable the **AJAXEvaluate** processor, in which case any API calls made to that processor are ignored.

<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable the <strong>AJAXEvaluate</strong> processor.</td>
<td>Disabled (API is disabled)</td>
</tr>
</tbody>
</table>

Apply ACLs to AJAXGlideRecord (client side Glide record)

From within client scripts, it is possible to query arbitrary data from the server via the **AJAXGlideRecord** (renamed to GlideAjax) API, by using a syntax similar to a server-side glide record. This is an extremely powerful and useful tool in many deployments.

If you choose to apply access control lists (ACL) to GlideAjax API calls, then you can only query data to which the currently connected user has rights to access. For example, if the user is logged in as an ESS user who has no rights to read the **cmn_location** table, then any GlideAjax API call on his behalf would fail.

If you run the system without an ACL checking on GlideAjax calls, then the API can return information that the currently logged in user could not otherwise access via the UI.

<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply standard security ACLs to AJAXGlideRecord calls</td>
<td>ACL checking enforced</td>
</tr>
</tbody>
</table>

Mutual authentication

Mutual Authentication establishes trust by exchanging secure sockets layer (SSL) certificates.

Before connecting to a server, the client requests an SSL certificate. The server responds by requesting that the client send its own certificate. Both respond by validating the certificates and sending acknowledgments before initiating an HTTPS connection.

Administrators do the preliminary work of setting up a keystore and generating certificates before certification requests are fulfilled.

**Warning:** This feature only enables mutual authentication on outbound https connections. The instance does not support mutual authentication on inbound requests.
Creating the Key Store

The instance currently supports uploading a Java keystore file to contain the private key, public certificate pair, and its signed certificates.

The following steps use commands that allow you to generate a new Java Keytool keystore file, create a certificate signing request (CSR), and import certificates. Any root or intermediate certificates need to be imported before importing the primary certificate for your domain. Type these commands in a command line interface.

1. Generate a Java keystore and key pair.
   
   ```
   keytool -genkey -alias mydomain -keyalg RSA -keystore my.keystore
   ```

2. Generate a CSR for an existing Java keystore.
   
   ```
   keytool -certreq -alias mydomain -keystore my.keystore -file mydomain.csr
   ```

3. Import a root or intermediate certificate authority CA certificate to an existing Java keystore.
   
   ```
   keytool -import -trustcacerts -alias root -file Thawte.crt -keystore my.keystore
   ```

4. Import a signed primary certificate to an existing Java keystore.
   
   ```
   keytool -import -trustcacerts -alias mydomain -file mydomain.crt -keystore my.keystore
   ```

Setting up the Key Store

Now that the key store has been created, it can be uploaded to the Certificates table. On the System Definition > Certificates page, click New and set the following fields:

- Enter a certificate Name.
- Store the key store as Active.
- Set Type = Java Key Store.
- Provide a Key store password. This is the password that was used to create the keystore.

Click Submit to create the Java Key Store entry.

![Keystore Table](image)

Figure 683: Keystore
Specifying a Trusted Server Certificate

During an outbound SSL connection, which is an HTTPS Web Service call, it is possible to specify a certificate provided by the service provider that ensures the validity of the service provider during the SSL connection. For example, a browser attempting to connect to a secure service which identifies itself by a certificate.

By uploading the trusted server certificate, ServiceNow ensures that the service it is connecting to is valid and secure.

Create a new Certificate entry with the type of "Trust Store Cert" and attach a DER formatted certificate, or copy and paste its PEM format into the PEM Certificate field.

![Certificate Entry](image)

**Figure 684: Trusted Certificate**
### Processing Mutual Authentication Requests

**Figure 685: Certificate Exchange**

- When a client requests the server certificate for authentication, a certificate signing request (CSR) is generated.
- To respond to a CSR, the server generates two unique cryptographic keys: A public key, which is used to encrypt messages to the server and a private key, which is used to decrypt messages. Both keys are kept in the Key Store.
- Keys are used to decrypt the client secure messages so they can be read by the server. Any outgoing connection that is going to be HTTPS verifies the certification by checking the Key Store, offering its public certification, and uses the trust store certificates to verify mutual trust back.
- To complete the secure link between the client and the server, the server matches the certificate to the corresponding private key. Because only the server has access to the private key, the server can decrypt the data from the client.

Here is an example of a command that registers MYHTTPS with the `com.glide.certificates.DBKeyStoreSocketFactory` socket factory on port 443. The database key store factory is used during the SSL exchange process to offer a client certificate for mutual authentication.

```java
import com.glide.certificates.DBKeyStoreSocketFactory;
glide.httpclient.protocol.myhttps.class = "com.glide.certificates.DBKeyStoreSocketFactory"
glide.httpclient.protocol.myhttps.port = "4433"
```

Having the above configuration will affect any outbound `myhttps://host.domain.com/target` URL to use the custom socket factory and exchange certificates during SSL.

**Note:** if you override the default 'https' protocol socket factory, you will be doing that on a global scale and it will affect every outbound https connection. This is usually undesirable.

The server responds by sending a certificate. Is this a certificate that the client accepts? If yes, a message is sent to the server accepting the certificate and a secure channel is initiated. If the certificate is not accepted, it may mean that the root authority is needed for certification.
Web service security

Web service security is enforced using the combination of basic authentication challenge/response for the HTTP protocol, as well as system level access control using the Contextual Security Manager. Additionally, there is a set of web service specific roles that may be granted to the web service user.

Basic Authentication

To enforce basic authentication on each request for a WSDL document or posting of SOAP messages, you may set the property glide.basicauth.required to `true`. If you do so, each WSDL or SOAP request would have to contain the "Authorization" header as specified in the Basic Authentication protocol. Because the request is non-interactive, we always require the Authorization header during a request.

Supplying basic authentication information whether or not it is required has the added advantage that the data created or updated as a result of the Web Service invocation is done on behalf of the user supplied in the basic authentication credentials. As an example, when creating an Incident record, the journal fields will have the user id of the basic authenticated user, instead of the default Guest user.

To make the authorization header ignore the capitalization rules, use the glide.security.script.include.name.case.insensitive.list property. You can modify this property in the System Properties [sys_properties] table and add the script includes that are necessary to process the authentication. By default, this property has these values: BasicAuth,CustomAuth for those two script includes. Add other script includes as needed. This property is available starting with the Geneva release.

To supply basic authentication when using Perl and the SOAP::Lite libraries, you can implement the following function:

```perl
sub SOAP :: Transport :: HTTP :: Client :: get_basic_credentials { return 'user_name' => 'password'; }
```

• When using C# .NET VS 2005 or older, you can take advantage of the Credentials object, for example:

```csharp

service . ServiceNow proxy = new service . ServiceNow ( ) ;
service . get getService = newservice . get ( ) ;
service . getResponse getServiceResponse = new service . getResponse ( ) ;

try {
    proxy . Credentials = cred ;
    getService . sys_id = "bf522c350a0a140701972dbf876f1610" ;
    getServiceResponse = proxy . get (getService ) ;
    catch (Exception ex) {
}
```

• When using C# .NET VS 2008, you can take advantage of the ClientCredentials object, for example:

```csharp
Demo_Incident. ServiceNowSoapClient client = new Test08WebService . Demo_Incident . ServiceNowSoapClient ( ) ;
client . ClientCredentials . UserName . UserName = "admin" ;
client . ClientCredentials . UserName . Password = "admin" ;
```

Then in your app.config file look for the following and change "None" to "Basic":

```xml
<transport clientCredentialType= "None" proxyCredentialType= "None" realm= "" />
```
• When using VB .NET taking advantage of the Credentials object would look like the following:

```vbnet
Sub Main()
    Dim cred As New System.Net.NetworkCredential( "user_name", "password"")
    Dim proxy As New VB_Democm.incident.ServiceNow
    Dim getIncident As New VB_Democm.incident.get
    Dim getResponse As New VB_Democm.incident.getResponse
    proxy.Credentials = cred
    getIncident.sys_id = "[your sysID here]"
    getResponse = proxy.get(getIncident)
End Sub
```

The resulting response when Basic Authentication is turned on and no credentials are supplied looks like this:

```html
<html> <head> <title>Apache Tomcat/5.0.28 - Error report </title> </head> <body> <h1>HTTP Status 401 - </h1> <p><b>type</b> Status report </p> <p><b>message</b> <u>This request requires HTTP authentication ().</u> </p> <p><b>description</b> <u>This request requires HTTP authentication ().</u> </p> <p><b>headers</b> </p> <h3>Apache Tomcat/5.0.28</h3> </body> </html>
```

WS-security

Support for WS-Security 1.1 in the form of WSS X.509 Token Profile and WSS Username Token Profile is available for incoming SOAP requests.

The configuration to use WS-Security is separate from the requirement to enforce Basic Authentication, and is enforced when the SOAP envelope contains the WS-Security headers. The following **WS Security Profiles** module contains configurations for using this feature.
The **WS Security Profile** module lists the WS-Security profiles that are currently in effect. The **Order** of the profiles indicate the order of authentication that will be checked, all profiles will be checked during the incoming SOAP request, when a profile fails authentication, it will not execute the next one in order. The **Bind session** checkbox indicates which profile to use to assume the session's identity, there can only be one "bound" session.

![WS Security Profiles Table]

**Figure 687: WS Security List**

**WSS X.509 Token Profile**

Use the X.509 authentication framework as defined by the Web Services Security: SOAP Message Security specification. An X.509 certificate specifies a binding between a public key and a set of attributes that includes (at least) a subject name, issuer name, serial number and validity interval. An X.509 certificate is used to validate a public key that is used to sign the incoming SOAP message. Upload the
certificate in the **Certificate** module and reference it in the **X509 Certificate** field. If this is a bound session, select the user to impersonate when the WS-Security authentication succeeds.

See the following document: [http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf](http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf)

Figure 688: WS x509

**WSS Username Token Profile**

In addition to specifying the X.509 Token Profile, a UsernameToken can also be supplied in the SOAP request. A UsernameToken is used as a means of identifying the requester by "username", and optionally using a password (or shared secret, or password equivalent) to authenticate that identity to the instance. The UsernameToken profile cannot be used independent of the X.509 Token Profile currently.

1. **Authenticate using the Username of the incoming SOAP request to lookup a User by the specified User field to match UserName** value. The password value in the incoming Username Token is used to authenticate the request. When the **Bind session** option is selected, the user that authenticates successfully will be used for the session.

   ![WS Security Profile](image)

   **Figure 689: Selecting Bind session**

2. **Authenticate using a separate pair of user name / password that is unrelated to users in the User table.** When the **Bind session** option is selected, the user that is specified in the **Run as user** field will be used for the session.

**Example WS-Security SOAP Envelope Headers**

---

**Note:**
This sample has been formatted with line returns to fit the content into the frame.

```xml
```
Set security for items and categories

Administrators can control access to content in the service catalog by creating and applying user criteria records.

Catalog managers and catalog editors can apply existing user criteria for items and categories to which they are assigned. However, catalog managers and catalog editors cannot create or edit user criteria directly.

You can **create user criteria records** that define conditions for user information. Then apply these criteria records to catalog items and categories, controlling access to these items and categories.

For example, create a **USA Sales** user criteria record matching users who are both in the sales team and based in the USA. Then apply this record to the **USA IT Hardware** catalog category, so only users matching the record can access that category.

The feature is active by default in new Fuji instances and upgraded instances that do not use entitlement-based access controls. If you upgrade an instance that uses entitlements, you can **migrate to user criteria** to take advantage of the improved control, flexibility, and reuse.

Access controls allow you to:

- Manage access to multiple items and categories in one user criteria record. For example, create a single **UK Employees** user criteria record and apply it to multiple items and categories that are available to employees in the UK.

© 2017 ServiceNow. All rights reserved. 2468
• Allow access if one condition matches, or if all conditions match. For example, define that only users who are both in a specific location and who belong to a specific department can have access.

• Use **Available For** and **Not Available For** lists to allow or prohibit access to users matching the conditions in a criteria record. For example, specify that a US-based catalog category is available for the users located in the USA but is not available for users belonging to the Sales department (whether in the US or not).

• **Extend matching** to create conditions matching additional fields in the User [sys_user] table, without having to use a script. For example, you can add a condition to match items against the **Cost center** field in user records.

**Note:** For changes in User [sys_user] table records to be effective, you should log out and log in.

Create a user criteria record

User criteria records define conditions that are evaluated against user records.

You can apply several user criteria records to a single catalog item or category. In this situation, users need to match only one of these criteria records to have access.

**Note:** When a change is made to the end user profile such as their location so that it affects their ability to view items, then those changes will not take effect till the end user has relaunched their session.

1. Navigate to **Service Catalog > Catalog Definition > User Criteria** and create a record.

2. Fill in the fields on the form, as appropriate.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the criteria record.</td>
</tr>
<tr>
<td>Users</td>
<td>The individual user records to match.</td>
</tr>
<tr>
<td>Groups</td>
<td>The group records to match.</td>
</tr>
<tr>
<td>Roles</td>
<td>The roles to match.</td>
</tr>
<tr>
<td>Advanced</td>
<td>A check box to display or hide the <strong>Script</strong> field.</td>
</tr>
<tr>
<td>Script</td>
<td>A <em>script</em> to define any additional criteria, and return <strong>true</strong> or <strong>false</strong>. This field is available only if <strong>Advanced</strong> is selected.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Because scripts are evaluated dynamically, including scripts in user criteria records can decrease performance.</td>
</tr>
<tr>
<td>Check box</td>
<td>For check boxes, describe the selected condition. For example: &quot;Check box for enabling the feature&quot; or &quot;Select the check box to enable the feature.&quot; Describe the cleared condition only when it is not obvious.</td>
</tr>
<tr>
<td>Active</td>
<td>A check box to activate or deactivate this criteria record.</td>
</tr>
<tr>
<td>Companies</td>
<td>The companies to match.</td>
</tr>
<tr>
<td>Locations</td>
<td>The locations to match.</td>
</tr>
<tr>
<td>Departments</td>
<td>The departments to match.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
**Match All** | A check box to determine whether all elements from each populated criteria field must match. If selected, only users who match all criteria are given access. If cleared, the user must meet one or more of the set criteria to be given access.

By default, this check box is cleared so that any condition met provides a match.

For example, consider a user criteria record for the following:
- Locations A or B
- Company C or D

With **Match All** selected, only users meeting all of these conditions are matched. For example, a user with a location A and a company C.

With **Match All** cleared, users meeting any of these conditions are matched. For example, a user with a location B.

**Note:** If you select **Match All**, ensure that you do not create contradictory conditions which can never be met. For example, if all users in location A work for company G, then the conditions in this example can never be met.

### Apply user criteria to items and categories

You can apply a user criteria record to items and categories, either from the item or category form or from the user criteria form.

**Note:** The user criteria restriction applies only within Service Catalog for the specific item it is applied for. However, the user criteria restriction is not applied outside the Service Catalog where the user has access to the item via the table.

To apply criteria directly to an item or category, use the **Available For** or **Not Available For** related lists in the Item or Category form.

**Note:**
- When a change is made to the end-user profile, such as location, and it affects their ability to view items, those changes do not take effect until the end user has relaunched the session.
- The user criteria restrictions of a category do not automatically apply to the catalog items within the category. If required, you should apply the user criteria restrictions to the individual catalog items.
To apply criteria directly from the User Criteria form, add items to the **Available For Catalog Items** and **Not Available For Catalog Items** related lists, and add categories to the **Available For Categories** and **Not Available For Categories** related lists. You may need to configure the User Criteria form to add these lists.

**Note:**
The **Not Available For** settings override **Available For** settings. A user on the **Not Available For** list for an item cannot access that item, even if that user is also on the **Available For** list for that item.

---

**Implement user criteria best practice**

When creating user criteria for your system, consider the following good practice recommendations.

- **Design Criteria for Reuse:**
  Ensure that you design user criteria for maximum reuse. Create user criteria records with common sets of conditions matching your organization’s requirements. Allow them to be shared across as many items and categories as possible, rather than creating multiple similar criteria records for individual items and categories.

- **Naming conventions:** Give each user criteria record a meaningful name, to help you determine the function for that record. For example,
  - Users in company Cloud Dimensions AND in London
  - Users in company Cloud Dimensions OR in London
  - Users belonging to the Group Development, IT, or Sales
  - Users with role itil, asset_manager, or catalog_admin

- **Test user criteria on a non-production instance:** Consider testing user criteria on a development or test instance, and then transferring the records from the **user criteria tables** and catalog records to your production instance using **update sets**.

---

**Extend user criteria**

Service catalog enables you to extend user criteria to match against additional reference fields in the User [sys_user] table, such as hidden or custom fields, without using a script.

You can then add a matching field in the User Criteria [user_criteria] table, allowing you to define conditions based on this field in user criteria records.

**Note:** The column on the User table must be a reference field, and the matching column on the User Criteria table must be a glide_list type field. Also, both columns must have matching names. Columns with a “U” prefix are catered for, so, for example, “cost_center” and “u_cost_center” are considered as matching.

1. Navigate to **System Definition > Tables**.
2. Locate and open the **User Criteria** record.
3. In the **Columns** section, insert a row for the new field.
4. Set the type to **List** and select the appropriate reference field. For example, select **cmn_cost_center** to set criteria based on cost centers.
   You have a step with a list, note, and image.
5. Navigate to **Service Catalog > Catalog Definition > User Criteria**.
6. Select or create a new user criteria record, and see that you can now use the new field.
You can further extend the User Criteria table to match against any columns in other tables. In the User [sys_user] table, add a new reference field to the other table, then extend the user criteria table to refer to that reference field, as described above.

Useful security scripts

There are several useful scripts you can use to secure your instance.

For an easy-to-navigate version, visit the Useful Scripts portal.

Caution: The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

Sample Java BouncyCastle Algorithm for Encryption

```java
// Java imports
import java.io.BufferedReader;
import java.io.ByteArrayOutputStream;
import java.io.ByteArrayInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.security.NoSuchProviderException;
import java.security.SecureRandom;
import java.security.Security;
import java.util.Iterator;
import sun.misc.BASE64Encoder;
import sun.misc.BASE64Decoder;

// Bouncy castle imports
import org.bouncycastle.jce.provider.BouncyCastleProvider;
import org.bouncycastle.openpgp.PGPCompressedData;
import org.bouncycastle.openpgp.PGPCompressedDataGenerator;
import org.bouncycastle.openpgp.PGPCompressedDataList;
import org.bouncycastle.openpgp.PGPException;
import org.bouncycastle.openpgp.PGP LiteralData;
import org.bouncycastle.openpgp.PGPObjectFactory;
import org.bouncycastle.openpgp.PGPSecretKey;
import org.bouncycastle.openpgp.PGPSecretKeyRingCollection;
import org.bouncycastle.openpgp.PGPUtil;

// public class SingleSignOnTest
private static File publicKeyFile = new File("/Development/Java/Single Sign On with Encryption(PGP)/PGP1D0.pkr");
private static File privateKeyFile = new File("/Development/Java/Single Sign On with Encryption(PGP)/PGP1D0.skr");
private static String privateKeyPassword = "passphrase";

// Public class method decrypt
public static String decrypt(byte[] encdata) {
    System.out.println("decrypt(): data length="+encdata.length);
    // ----- Decrypt the file
    try {
        ByteArrayInputStream bais = new ByteArrayInputStream(encdata);
        FileInputStream privKey = new FileInputStream(privateKeyFile);
        return _decrypt(bais, privKey, privateKeyPassword.toCharArray());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        e.printStackTrace();
        return null;
    }
}

// Public class method encrypt
public static byte[] encrypt(byte[] data) {
    // ----- Read in the public key
    FileInputStream privKey = new FileInputStream(privateKeyFile);
    return _encrypt(bais, privKey, privateKeyPassword.toCharArray(), toCharArray());
}
```

© 2017 ServiceNow. All rights reserved. 2473
PGPPublicKey key = readPublicKeyFromCol(new FileInputStream(publicKeyFile));
System.out.println("Creating a temp file..."); // create a file and write the string to it
File tempfile = File.createTempFile("pgp", null);
FileOutputStream fos = new FileOutputStream(tempfile);
fos.write(data);
fos.close();
System.out.println("Temp file created at ");
System.out.println(tempfile.getAbsolutePath());
System.out.println("Reading the temp file to make sure that the bits were written
-----
BufferedReader isr = new BufferedReader(new FileReader(tempfile));
String line = "";
while ((line = isr.readLine) != null) {
    System.out.println(line + "\n");
}
// find out a little about the keys in the public key ring
getBitStrength();
getAlgorithm();
getBitStrength();
System.out.println("Version = " + key.getVersion());
isEncryptionKey() + ", Master key = " + key.isMasterKey());
int count = 0;
for (java.util.Iterator iterator = key.getUserIDs(); iterator.hasNext(); ) {
    count ++;
    System.out.println((String) iterator.next());
}
System.out.println("Key Count = " + count);
// create an armored ascii file
FileOutputStream out = new FileOutputStream(outputfile); // encrypt the file
encryptFile(tempfile.getAbsolutePath(), out, key); // Encrypt the data
ByteArrayOutputStream baos = new ByteArrayOutputStream();
_encrypt(tempfile.getAbsolutePath(), baos, key);
System.out.println("encrypted text length=");
baos.size();
tempfile.delete(); return baos.toByteArray();
catch (PGPEXception e) { // System.out.println(e.toString());
    System.out.println(e.getUnderlyingException().toString());
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
Private class method readPublicKeyFromCol // private static PGPPublicKey
readPublicKeyFromCol(InputStream in) throws Exception {
    PGPPublicKeyRing pkRing = null;
    PGPPublicKeyRingCollection pkCol = new PGPPublicKeyRingCollection(in);
    System.out.println("key ring size=" + pkCol.size());
    Iterator it = pkCol.getKeyRings(); while (it.hasNext) {
        pkRing = (PGPPublicKeyRing) it.next();
        Iterator pkIt = pkRing.getPublicKeys(); while (pkIt.hasNext) {
PGPPublicKey key = (PGPPublicKey) pkIt.next();
System.out.println("Encryption key = "+
key.isEncryptionKey()+ ", Master key = "+
key.isMasterKey());
if (key.isEncryptionKey()) return key; // Private
class method _encrypt // private static void _encrypt (String fileName, OutputStream out, PGPPublicKey encKey)
throws IOException, NoSuchProviderException, PGPEXception
{
    out = new DataOutputStream(out);
    ByteArrayOutputStream bOut = new ByteArrayOutputStream();
    System.out.println("creating comData... ");
    // get the data from the original file
    PGPCompressedDataGenerator comData = new
    PGPCompressedDataGenerator(PGPCompressedDataGenerator.ZIP);
    PGPUtility.writeFileToLiteralData(comData.open(bOut),
    PGPLiteralData.BINARY, new File(fileName));
    comData.close();
    System.out.println("comData created... ");
    System.out.println("using PGPEncryptedDataGenerator... ");
    // object that encrypts the data
    PGPEncryptedDataGenerator cPk = new PGPEncryptedDataGenerator
    (PGPEncryptedDataGenerator.CAST5, new SecureRandom(), "BC");
    cPk.addMethod(encKey);
    System.out.println("used PGPEncryptedDataGenerator...");
    // take the outputstream of the original file
    byte[] bytes = bOut.toByteArray();
    System.out.println("wrote bOut to byte array...");
    // write the plain text bytes to the armored outputstream
    OutputStream cOut = cPk.open(out, bytes.length);
    cOut.write(bytes);
    cPk.close();
    out.close();
}
private static String _decrypt (InputStream in, InputStream keyIn, char[] passwd) throws Exception {
    in = PGPUtility.getDecoderStream(in);
    try {
        PGPObjectFactory pgpF = new PGPObjectFactory(in);
        PGPEncryptedDataList enc;
        Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
        PGPEncryptedDataList) {
            enc = (PGPEncryptedDataList) o; } else {
                enc = (PGPEncryptedDataList) pgpF;
            }
        PGPEncryptedDataList enc;
        Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
        PGPEncryptedDataList) {
            enc = (PGPEncryptedDataList) o; } else {
                enc = (PGPEncryptedDataList) pgpF;
            }
        PGPObjectFactory pgpF = new PGPObjectFactory(in);
        Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
        PGPEncryptedDataList) {
            enc = (PGPEncryptedDataList) o; } else {
                enc = (PGPEncryptedDataList) pgpF;
            }
        if (o instanceof PGPCompressedData) {
            PGPCompressedDataGenerator comData = new
            PGPCompressedDataGenerator(PGPCompressedDataGenerator.ZIP);
            PGPUtility.writeFileToLiteralData(comData.open(bOut),
            PGPLiteralData.BINARY, new File(fileName));
            comData.close();
            System.out.println("comData created... ");
            System.out.println("using PGPEncryptedDataGenerator... ");
            // object that encrypts the data
            PGPEncryptedDataGenerator cPk = new PGPEncryptedDataGenerator
            (PGPEncryptedDataGenerator.CAST5, new SecureRandom(), "BC");
            cPk.addMethod(encKey);
            System.out.println("used PGPEncryptedDataGenerator...");
            // take the outputstream of the original file
            byte[] bytes = bOut.toByteArray();
            System.out.println("wrote bOut to byte array...");
            // write the plain text bytes to the armored outputstream
            OutputStream cOut = cPk.open(out, bytes.length);
            cOut.write(bytes);
            cPk.close();
            out.close();
        } // // Private class method _decrypt //
        private static String _decrypt (InputStream in, InputStream keyIn, char[] passwd) throws Exception {
            in = PGPUtility.getDecoderStream(in);
            try {
                PGPObjectFactory pgpF = new PGPObjectFactory(in);
                PGPEncryptedDataList enc;
                Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
                PGPEncryptedDataList) {
                    enc = (PGPEncryptedDataList) o; } else {
                        enc = (PGPEncryptedDataList) pgpF;
                    }
                PGPObjectFactory pgpF = new PGPObjectFactory(in);
                Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
                PGPEncryptedDataList) {
                    enc = (PGPEncryptedDataList) o; } else {
                        enc = (PGPEncryptedDataList) pgpF;
                    }
                if (o instanceof PGPCompressedData) {
                    PGPCompressedDataGenerator comData = new
                    PGPCompressedDataGenerator(PGPCompressedDataGenerator.ZIP);
                    PGPUtility.writeFileToLiteralData(comData.open(bOut),
                    PGPLiteralData.BINARY, new File(fileName));
                    comData.close();
                    System.out.println("comData created... ");
                    System.out.println("using PGPEncryptedDataGenerator... ");
                    // object that encrypts the data
                    PGPEncryptedDataGenerator cPk = new PGPEncryptedDataGenerator
                    (PGPEncryptedDataGenerator.CAST5, new SecureRandom(), "BC");
                    cPk.addMethod(encKey);
                    System.out.println("used PGPEncryptedDataGenerator...");
                    // take the outputstream of the original file
                    byte[] bytes = bOut.toByteArray();
                    System.out.println("wrote bOut to byte array...");
                    // write the plain text bytes to the armored outputstream
                    OutputStream cOut = cPk.open(out, bytes.length);
                    cOut.write(bytes);
                    cPk.close();
                    out.close();
                } // // Private class method _decrypt //
                private static String _decrypt (InputStream in, InputStream keyIn, char[] passwd) throws Exception {
                    in = PGPUtility.getDecoderStream(in);
                    try {
                        PGPObjectFactory pgpF = new PGPObjectFactory(in);
                        PGPEncryptedDataList enc;
                        Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
                            PGPEncryptedDataList) {
                                enc = (PGPEncryptedDataList) o; } else {
                                    enc = (PGPEncryptedDataList) pgpF;
                                }
                            PGPObjectFactory pgpF = new PGPObjectFactory(in);
                            Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
                                PGPEncryptedDataList) {
                                    enc = (PGPEncryptedDataList) o; } else {
                                        enc = (PGPEncryptedDataList) pgpF;
                                    }
                                if (o instanceof PGPCompressedData) {
                                    PGPCompressedDataGenerator comData = new
                                    PGPCompressedDataGenerator(PGPCompressedDataGenerator.ZIP);
                                    PGPUtility.writeFileToLiteralData(comData.open(bOut),
                                    PGPLiteralData.BINARY, new File(fileName));
                                    comData.close();
                                    System.out.println("comData created... ");
                                    System.out.println("using PGPEncryptedDataGenerator... ");
                                    // object that encrypts the data
                                    PGPEncryptedDataGenerator cPk = new PGPEncryptedDataGenerator
                                    (PGPEncryptedDataGenerator.CAST5, new SecureRandom(), "BC");
                                    cPk.addMethod(encKey);
                                    System.out.println("used PGPEncryptedDataGenerator...");
                                    // take the outputstream of the original file
                                    byte[] bytes = bOut.toByteArray();
                                    System.out.println("wrote bOut to byte array...");
                                    // write the plain text bytes to the armored outputstream
                                    OutputStream cOut = cPk.open(out, bytes.length);
                                    cOut.write(bytes);
                                    cPk.close();
                                    out.close();
                                } // // Private class method _decrypt //
                                private static String _decrypt (InputStream in, InputStream keyIn, char[] passwd) throws Exception {
                                    in = PGPUtility.getDecoderStream(in);
                                    try {
                                        PGPObjectFactory pgpF = new PGPObjectFactory(in);
                                        PGPEncryptedDataList enc;
                                        Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
                                            PGPEncryptedDataList) {
                                                enc = (PGPEncryptedDataList) o; } else {
                                                    enc = (PGPEncryptedDataList) pgpF;
                                                }
                                            PGPObjectFactory pgpF = new PGPObjectFactory(in);
                                            Object o = pgpF.nextObject(); // // the first object might be a PGP marker packet. // if (o instanceof
                                                PGPEncryptedDataList) {
                                                    enc = (PGPEncryptedDataList) o; } else {
                                                        enc = (PGPEncryptedDataList) pgpF;
                                                    }
                                                if (o instanceof PGPCompressedData) {
                                                    PGPCompressedDataGenerator comData = new
                                                    PGPCompressedDataGenerator(PGPCompressedDataGenerator.ZIP);
                                                    PGPUtility.writeFileToLiteralData(comData.open(bOut),
                                                    PGPLiteralData.BINARY, new File(fileName));
                                                    comData.close();
                                                    System.out.println("comData created... ");
                                                    System.out.println("using PGPEncryptedDataGenerator... ");
                                                    // object that encrypts the data
                                                    PGPEncryptedDataGenerator cPk = new PGPEncryptedDataGenerator
                                                    (PGPEncryptedDataGenerator.CAST5, new SecureRandom(), "BC");
                                                    cPk.addMethod(encKey);
                                                    System.out.println("used PGPEncryptedDataGenerator...");
                                                    // take the outputstream of the original file
                                                    byte[] bytes = bOut.toByteArray();
                                                    System.out.println("wrote bOut to byte array...");
                                                    // write the plain text bytes to the armored outputstream
                                                    OutputStream cOut = cPk.open(out, bytes.length);
                                                    cOut.write(bytes);
                                                    cPk.close();
                                                    out.close();
                                                } // // Private class method _decrypt //
Geneva    ServiceNow    ServiceNow Platform

PGPCompressedData cData =
(PGPCompressedData ) message;
PGPObjectFactory pgpFact = new
PGPObjectFactory (cData. getInputStream ( ) );
message = pgpFact. nextObject ( ) ;
ByteArrayOutputStream baos = new
ByteArrayOutputStream ( ) ;

if (message instanceof PGPLiteralData ) {
PGPLiteralData ld = (PGPLiteralData )
message ;
InputStream unc = ld. getInputStream ( ) ;
int ch ; while ( (ch = unc. read ( ) ) &gt ;= 0 ) {
    baos. write (ch ) ;
} else if
(message instanceof PGPOnePassSignatureList ) { throw new PGPException
("encrypted message contains a signed message - not literal data." ) ;
} else { throw new PGPException ( "message is not a simple encrypted file -
type unknown." ) ;
} if (pbe. isIntegrityProtected ( ) ) { if ( !pbe. verify ( ) ) {
    System. err. println ( "message failed integrity check" ) ;
} else { System. err. println ( "message integrity check passed" ) ;
} } else {
    System. err. println ( "no message integrity check" ) ;
} return baos. toString ( ) ;
}

catch (PGPException e ) {
    System. err. println (e ) ;
    if (e. getUnderlyingException ( ) != null ) {
        e. getUnderlyingException ( ). printStackTrace ( ) ;
    }
}
return null ;

// Private class method findSecretKey //
private static PGPPrivateKey findSecretKey (InputStream keyIn , long
keyID ,
    char [ ] pass ) throws IOException , PGPException ,
NoSuchProviderException ( 
    PGPSecretKeyRingCollection pgpSec = new
PGPSecretKeyRingCollection (PGPUtil. getDecoderStream (keyIn ) ) ;
    PGPSecretKey pgpSecKey = pgpSec. getSecretKey (keyID ) ;
if
(pgpSecKey == null ) { return null ;
} return pgpSecKey. extractPrivateKey
(pass , "BC" ) ;
} // // Public class method readFile //
public byte [ ] readFile (File file ) throws IOException  {
FileInputStream fis = new FileInputStream (file ) ;
byte [ ] buf = new byte [ 4096 ] ;
int numRead = 0 ;
ByteArrayOutputStream baos = new ByteArrayOutputStream ( ) ;

while ( (numRead = fis. read (buf ) ) &gt ; 0 ) {
    baos. write (buf , 0 , numRead ) ;
}
    byte [ ] returnVal = baos. toByteArray ( ) ;
    baos. close ( ) ;
    return returnVal ;
} // Public main method //
public static  void main (String [ ] args ) {

Security. addProvider ( new BouncyCastleProvider ( ) ) ;
String TOKEN = "aamine" ; // ----- Encrypt the message to a
file //

byte [ ] enccdata = encrypt (TOKEN. getBytes ( ) ) ;
System. out. println ( "Encrypted: " + enccdata ) ;

the byte array to a string
BASE64Encoder en = new BASE64Encoder ( ) ;
String temp = en. encode (enccdata ) ;
System. out. println ( "Temp: " + temp ) ;

byte [ ] newB = null ;
BASE64Decoder en1 = new BASE64Decoder ( ) ;
try {
    newB = enl. decodeBuffer (temp ) ;
} catch (Exception e ) {
    System. out. println ( "Exception: " + e ) ;
}
System. out. println ( "byte array" + newB. length ) ;

----- Decrypt the token that
Set Write Role for All Catalog Variables

This script sets a write role of Admin for all catalog variables in the system.

Name: Set Write Role for All Catalog Variables

Type: 

Table: 

Description: This script sets a write role of Admin for all catalog variables in the system.

Parameters: 

Script:

```javascript
function pat () {
    var gr = new GlideRecord ( 'item_option_new' ) ; var count = 0 ;
    gr. query () ; while (gr. next ( ) ) {
        count = count + 1 ;
        gr. setValue ( 'write_roles' , 'admin' ) ;
        gr. update ( ) ;
        gs. print ( "UPDATED ROLE # " + count ) ;
    }
}
```

Make a page public

Administrators can make pages public.

Role required: admin

⚠️ Warning: Several base-system public pages are required for the functionality of many features. Do not disable base-system public pages.

**Note:** Most pages are only viewable by logged in users. A limited number of pages are public so that users do not have to log in to view them, such as the welcome page, the front page, and the login and logout pages.

1. Navigate to **System Definition > Public Pages**.
2. Click **New**.
3. Specify the name of the page and click **Save**.

Make a public page private

Administrators can remove pages from public view.

Role required: admin

⚠️ Warning: Several base-system public pages are required for the functionality of many features. Do not disable base-system public pages.
1. Navigate to System Definition > Public Pages.
2. Locate the page name on the list.
3. Set the Active value to false. Users must now log in to access the page.

Activate the public pages module

The Public Pages module can be activated in the System Definition application.
Role required: admin

Warning: Several base-system public pages are required for the functionality of many features. Do not disable base-system public pages.

Note: Most pages are only viewable by logged in users. A limited number of pages are public so that users do not have to log in to view them, such as the welcome page, the front page, and the login and logout pages.

If the application menu does not show the Public Pages module in the System Definition application, you must activate it.

1. Navigate to System Definition > Application Menus.
2. Open the System Definition record.
3. In the Modules related list, click Public Pages.
4. In the Visibility section, click Active.
5. Click Update.

Login security

Login security refers to the security settings you can configure to control access to your instance.

Specify a login landing page

By default, users see their homepage upon login. You can specify a different login landing page by using a system property or the content management system.
Role required: admin

To specify a login landing page for all users, change the property value on the sys_properties table.

1. Type sys_properties.list in the navigation filter.
2. Locate the glide.login.home system property.
3. In the Value field, enter the name of the page that all users will see upon login.

Use <page name>.do; you may omit the http://''instance''.service-now.com/ portion of the URL. To determine the page name or the URL of a page in the system, you can point to a link. Some possible pages are welcome.do and incident.do.

Note: This property is system-wide, so setting it affects all users.
Enable the logout confirmation prompt

You can enable a logout confirmation prompt to prevent users from inadvertently logging themselves out.

Role required: admin

**Note:** The following procedure does not work in UI16.

1. Navigate to **System Properties > System**.
2. Locate the **Prompt user to confirm a logout request** property and select the check box.
3. When the user clicks the **Logout** button, a confirmation dialog box displays.

![Confirmation dialog box](image)

Remove the Logout button

You can remove the **Logout** button to prevent inadvertent logouts.

Role required: admin

**Note:** The following procedure does not work in UI16.

1. Navigate to **User Administration > User Preferences**.
2. Delete the system preference user.can.logout.

IP range based authentication

One way to secure a web-based application is to restrict access based on the IP address.

You can block access to a specific address or range of addresses that you suspect belong to malicious individuals. The instance allows you to control access by IP address.

Notes and Limitations:

- The system won't let you lock yourself out, so if you try to add a rule such that your current address would be locked out, the system will warn you and refuse your insert.
- If you're inside of a corporate intranet, be very careful about setting up your IP rules. The IP address you see on your own computer (like 10.10.10.25) generally bears no relationship to the IP address you'll actually appear as out on the internet. Your company will likely proxy and/or NAT your address into a predictable set of outbound addresses which you'll likely need to ask your network team about.
- A user whose access is restricted based on an access rule will get a 403 error on their browser.
- Restricted users don't use transactions, semaphores or count towards any server resource counts.
• This feature doesn't supersede or override your existing access control rules if, for example, you're running a VPN to our data center. It's an additional check that must be met in addition to any access controls we may have set up on your PIX.
• Allow rules always supersede deny rules. So if my address is both allowed (by one rule) and denied (by a second rule) it is, in fact, allowed.
• Asterisks and CIDR blocks are not currently supported.
• Regarding forwarded proxy addresses, the allow rules are applied to each address in the chain and then the deny rules are applied to each address in the chain if none of the allow rules matched.

Access control

By default the list is empty, meaning that there are no particular restrictions on access to your instance.

Role required: admin

Navigate to System Security > IP Address Access Control to see a list of your IP access controls. You may need to activate this module.

You can add these types of rules:
• Allow: any IP address in this range is allowed to connect to this instance.
• Deny: any IP address in this range is not allowed to connect to this instance unless it is listed in an allow rule.

Note: These rules also affect transferring update sets. To ensure that IP Address Access Control does not cause update sets to fail, add the target instance as an exception on the source instance.

Example 1: Block a particular range

An example of how to block a particular range.

Let's say we want to block a particular range of IPs, say 64.236.16.0 -> 64.236.16.255. Click "new" to add a new rule. Then fill it in as follows. Range Start and Range End must be specific IP addresses as seen in the examples, without asterisks or CIDR blocks.

![Access Control](image)

**Figure 690: Deny range**

Example 2: Block everyone except a particular range

An example of how to block everyone except a particular range.

Let's turn the problem around. So if an address is between 64.236.16.0 -> 64.236.16.255 we want to allow it connect, but we want to deny everybody else. To do this we're going to need two rules, one to allow that range, and a second to deny everybody else.

Click **New** and add a new rule. Then fill it in as follows.
Now let's add the deny rule.

Find denied IP addresses

Denied IP addresses are by default not viewable from the system logs. However, you can still find them in the instance's node log files.

Role required: admin

Log entries for blocked IP address appear as follows:


2. Browse the logs by criteria, such as time period and message.
3. You can also download log files when you know which log you are looking for, by navigating to System Logs > Utilities > Node Log File Download.

Implementing a nonce

You can implement a nonce to be used with single sign-on digest authentication.

To use a nonce with the unencrypted token or encrypted token methods of single sign on, the steps in this article will still apply with only a few minor changes.

Note: The nonce is used only for login requests, not for any other type of request. If the system receives a nonce value after login, the nonce is not consumed.
Benefits

The usage of a nonce prohibits a malicious user from performing a replay attack in order to log into your system.

Nonce process flow

When a customer has implemented the digested token Single Sign-on and wishes to add the security of a nonce, they follow a certain process flow.

1. A user logs into the customer's portal.
2. The customer generates the required SSO parameters and appends a random nonce to the end. For example, if the customer were forwarding the authentication response via the query string, it may look something like this:

   \[\text{SM\_USER=itil}\&\text{DE\_USER=V1QuWMmxSfBgfRS099X0cAjo5Q}\&\text{NONCE=1407743018}\]

The instance receives this request and retrieves the authentication variables. Before attempting to verify the integrity of the authentication response, the instance checks the nonce against an internal table (u_authentication_nonce) to verify that it does not yet exist. If the nonce does not exist within that table, the nonce is then added to the table and the authentication process is allowed to continue. However, if that nonce value already exists within the table, the authentication attempt is cancelled and an error code of failed_missing_requirement is returned, which will typically take the user back to the login page.

Implement a nonce

Implementing a nonce is fairly straightforward when these steps are followed:

- Create a system property called glide.authenticate.header.nonce_key and set its value to whatever variable name you're using for the nonce, such as NONCE or NCE.
- Create a new table called u_authentication_nonce. Add a field to the table called u_nonce.
- Go to System Properties > Installation Exits and create a new item called DigestSingleSignOnNonce which overrides ExternalAuthentication (see image below).

![Installation Exit](image)

- Add the following code to the script portion of the newly created DigestSingleSignOnNonce.

    ```javascript
    gs.include("PrototypeServer");
    ```
var DigestSingleSignOnNonce = Class.create();

DigestSingleSignOnNonce.prototype = {

  process : function() {

    var headerKey = GlideProperties.get("glide.authenticate.header.key", "SM_USER");
    var headerDigestKey = GlideProperties.get("glide.authenticate.header.encrypted_key", "DIGEST");
    var headerNonceKey = GlideProperties.get("glide.authenticate.header.nonce_key", "NCE");
    var fieldName = GlideProperties.get("glide.authenticate.header.value", "user_name");
    var fkey = GlideProperties.get("glide.authenticate.secret_key");

    // Look in the Headers
    var data = request.getHeader(headerKey);
    var encdata = request.getHeader(headerDigestKey);
    var nonce = request.getHeader(headerNonceKey);

    // If not, then check the URL Parameters
    if (data == null || encdata == null || nonce == null) {
      data = request.getParameter(headerKey);
      encdata = request.getParameter(headerDigestKey);
      nonce = request.getParameter(headerNonceKey);
    }

    // then maybe its a cookie
    if (data == null || encdata == null || nonce == null) {
      var cookies = request.getCookies();
      data = GlideCookieMan.getCookieValue(cookies, headerKey);
      encdata = GlideCookieMan.getCookieValue(cookies, headerDigestKey);
      nonce = GlideCookieMan.getCookieValue(cookies, headerNonceKey);
    }

    // if found run encryption
    if (data != null && encdata != null && nonce != null) {
      try {

        // Replace all spaces with plus(+)’s, converted in url
        encdata = encdata.replaceAll(' ', '+');

        // ----- Encrypt the username|nonce
        var key = this.getDigest(data + '|' + nonce, fkey);

        // Check for match of received encoded data
        // and your encoding of user name
        if (encdata == key) {
          var ugr = new GlideRecord("sys_user");
          ugr.initialize();
          if (!ugr.isValidField(fieldName)) {
            GlideLog.warn("External authorization is set to use field: ‘" + fieldName + "’ which doesn't exist");
            return "failed_missing_requirement";
          }
          ugr.addQuery(fieldName, data);
          ugr.query();
          if (!ugr.next()) {
            var userLoad = GlideUser.getUser(data);
            if (userLoad == null)
              return "failed_authentication";
          }
          ugr.initialize();
          ugr.addQuery(fieldName, data);

      } catch (e) {
        return "failed_encryption";
      }
    }
  }
}
ugr.query();
    if (!ugr.next())
        return "failed_authentication";
    }
    if (this.processNonce(nonce)){
        var userName = ugr.getValue("user_name");
        return userName;
    }
    else return "failed_missing_requirement";
    }
    else {
    return "failed_authentication";
    }
    }
    catch(e) {
    gs.log(e);
    return "failed_authentication";
    }
    // Encoded data didn't match recieved Encoded data
    }
    else {
    return "failed_missing_requirement";
    }
    }

    getDigest : function( data, fkey ) {
    try {
    // default to something JDK 1.4 has
    var MAC_ALG = "HmacSHA1";
    return SncAuthentication.encode(data, fkey, MAC_ALG);
    } catch (e) {
    gs.log(e.toString());
    throw 'failed_missing_requirement';
    }
    },

    processNonce : function( sentNonce ) {
    var ngr = new GlideRecord("u_authentication_nonce");
    ngr.addQuery("u_nonce", sentNonce);
    ngr.query();
    if (ngr.next()) {
    gs.log("This SSO entry has already been processed! (Nonce: " +
    sentNonce + ")");
    return false;
    }
    var ngrNew = new GlideRecord("u_authentication_nonce");
    ngrNew.initialize();
    ngrNew.u_nonce = sentNonce;
    ngrNew.insert();
    gs.log("Inserted new nonce: " + sentNonce);
    return true;
    };

    • Once you’ve saved your new installation exit, go to the DigestSingleSignOn installation exit and make
    sure that it is set Active=false.

    Your instance should now be configured to implement a nonce.
Installation exits

Installation exits are customizations that exit from Java to call a script before returning back to Java.

Note: Functionality described here requires the Admin role.

Available installation exits

Navigate to System Definition > Installation Exits. Some installation exit names (Login, Logout, ValidatePassword, ExternalAuthentication) are reserved and cannot be changed. Other installation exits can override these with custom script that replaces the script in the default installation exit.

The following installation exits are available in the base system:

<table>
<thead>
<tr>
<th>Installation Exit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>Takes a username and password pair and authenticates with the user object</td>
</tr>
<tr>
<td>Logout</td>
<td>Takes the user to the welcome page upon signing out; can be overridden by LogoutRedirect</td>
</tr>
<tr>
<td>LogoutRedirect</td>
<td>Takes the user to a specified URL upon signing out</td>
</tr>
<tr>
<td>ExternalAuthentication</td>
<td>Authenticates using header, parameter, or cookie; can be overridden by DigestSingleSignOn and PGPSingleSignOn</td>
</tr>
<tr>
<td>DigestSingleSignOn</td>
<td>Authenticates using header, parameter, or cookie and decrypts Digest encryption</td>
</tr>
<tr>
<td>PGPSingleSignOn</td>
<td>Authenticates using header, parameter, or cookie and decrypts PGP encryption</td>
</tr>
<tr>
<td>ValidatePassword</td>
<td>Inactive by default; allows customers to define their own password validation; can be overridden by ValidatePasswordStronger</td>
</tr>
<tr>
<td>ValidatePasswordStronger</td>
<td>Requires passwords be at least 8 characters long and contain a digit, an uppercase letter, and a lowercase letter</td>
</tr>
</tbody>
</table>

Login modifications

The following modification to the Login installation exit sets each user’s session timeout value as the user is logging in. In this particular example, if the user name is admin, the session is set to timeout in 30 seconds.

gs.include("PrototypeServer");
var Login = Class.create();
Login.prototype = {
    initialize : function() {
    },
}
process : function() {
    // the request is passed in as a global
    var userName = request.getParameter("user_name");
    var userPassword = request.getParameter("user_password");

    var authed = GlideUser.authenticate(userName, userPassword);
    if (authed) {
        // *********************************************************
        // customization - if the userName == admin, set the session
        // timeout to be 30 seconds. You can implement your own
        // session timeout algorithm here by checking to see if a user
        // belongs to a certain group or has a certain role.

        if (userName == "admin") {
            request.getSession().setMaxInactiveInterval(30);
        }
        // *********************************************************

        return GlideUser.getUser(userName);
    }

    this.loginFailed();
    return "login.failed";
},

loginFailed : function() {
    var message = GlideSysMessage.format("login_invalid");
    var gSession = GlideSession.get();
    gSession.addErrorMessage(message);
    var userName = request.getParameter("user_name");
    EventManager.queue("login.failed", ",", userName, ",");
}
}

Session timeout can also be set according to IP address.

gs.include("PrototypeServer");

var Login = Class.create();
Login.prototype = {
    initialize : function() {
    },

    process : function() {
        // the request is passed in as a global
        var userName = request.getParameter("user_name");
        var userPassword = request.getParameter("user_password");

        var authed = GlideUser.authenticate(userName, userPassword);
        if (authed) {

            // customization - if the user is logging in from a particular IP
            // range starting with XXX.XXX you can implement your own
            // session timeout algorithm here by checking the login IP

            var clientIP = gs.getSession().getClientIP().toString();
            // ...
        }
    }
}
// if client IP starts with specified range
if (clientIP.indexOf('XXX.XXX') == 0) {
    // set to 10 hours
    request.getSession().setMaxInactiveInterval(60 * 60 * 10);
}
// ***************************************************************
return GlideUser.getUser(userName);
}

this.loginFailed();

return "login.failed";
);

loginFailed : function() {
    var message = GlideSysMessage.format("login_invalid");
    var gSession = GlideSession.get();
    gSession.addErrorMessage(message);

    var userName = request.getParameter("user_name");
    EventManager.queue("login.failed", "", userName, "");

}

Strengthening password validation rules

You can customize password strength validation rules for the change password screen by overriding the installation exit associated with password validation.

1. Navigate to **System Definition > Installation Exits**.
2. Locate **ValidatePassword** and **ValidatePasswordStronger**. Both of these are inactive.
3. The **ValidatePasswordStronger** script (below) is a sample script that overrides the **ValidatePassword** script by using regular expressions to require that passwords be a minimum of 8 characters long, contain a numeric digit, and contain mixed-case letters.

```javascript
gs.include("PrototypeServer");
var ValidatePasswordStronger = Class.create();
ValidatePasswordStronger.prototype = {
    process : function() {
        var user_password = request.getParameter("user_password");
        var min_len = 8;
        var rules = "Password must be at least " + min_len + " characters long and contain a digit, an uppercase letter, and a lowercase letter.";
        if (user_password.length() < min_len) {
            gs.addErrorMessage("TOO SHORT: " + rules);
            return false;
        }
        var digit_pattern = new RegExp("[0-9]", "g");
        if (!digit_pattern.test(user_password)) {
            gs.addErrorMessage("DIGIT MISSING: " + rules);
            return false;
        }
        var upper_pattern = new RegExp("[A-Z]", "g");
        if (!upper_pattern.test(user_password)) {
            gs.addErrorMessage("UPPERCASE MISSING: " + rules);
            return false;
        }
        var lower_pattern = new RegExp("[a-z]", "g");
```

© 2017 ServiceNow. All rights reserved. 2487
The script variable created by `Class.create()` must have the same name as the installation exit itself – "ValidatePasswordStronger" in this example. The script implements the `process()` function which returns true if the password is acceptable and false if the password must be revised. The `gs.addErrorMessage` function can be used to return error messages on the change password screen. You can try this Installation Exit in your instance by checking the active flag and updating the record. Be sure and clear the cache after doing this so the change is recognized.

Also, keep in mind that modifying these scripts will not change the default ServiceNow behavior: Blank passwords are still prohibited by default and the password and verify password fields must match.

To test, check the **Password needs reset** box on a user record then login with that user. Validation will occur at the point that the user attempts to set the password. Validation does not apply when an admin user updates the password in the user record directly (the admin can put anything in the password field).

---

**Note:** The change password screen only applies to customers who do not use single sign on and are not integrated with their local LDAP.

---

### Login scenarios

Describes different login scenarios.

**Role required:** admin

When users log on to an instance directly, such as going to `http://{instance_name}.service-now.com/`, the system does the following:

1. Accesses the value in the property `glide.entry.page.script`. The default value of the property is derived from a script include named `CMSEntryPage`.
2. Directs the user to the instance login page if the entry page requires a login.
3. Applies login rules, if any, to the user.

To force the system to direct all users to the same page after login:

1. Navigate to **Content Management > Configuration > Configuration Page**.
2. Select a value for the Login page field, or create a new page as desired. If this page is not the site default page, it always redirects here. If it is a site default page, it applies login rules. If this value is null, the system uses `navpage.do` as the entry page. Do not enter a login page here; otherwise, users will need to log in twice.

**Logging Into an Instance to Access a Record:**

When users log into an instance to access a record by its globally unique identifier (sys_id), such as `http://{instance}.service-now.com/incident.do?sys_id={sys_id}`, then the system does the following:

1. Directs the user to a login page if not already logged in.
2. Directs the user to the appropriate record if they are allowed to access it. If the user does not have access rights to the record, a denial of access message appears.

**Logging Into a CMS Site:**
When users log on to a CMS site, such as http://<instance>.service-now.com/site-name/page.do, the system does the following:

- If there is a value in the Login page field on the CMS site form, it directs the user to that login page and applies login rules, if any, to the user.
- If there is no login page specified, it directs the user to the value in the Home page field on the CMS site form.

Logging Into a CMS Site to Access a Record:

When users log on to a CMS site to access a record, such as http://{instance}.service-now.com/ess/incident_detail.do?sysparm_document_key=incident,{sys_id}, the system follows the same procedure and finally takes the user to the record. If the user does not have access rights to the record, a denial of access message appears.

Logins and the employee self-service portal

The system keeps track of the first starting page that a user is trying to access even if the user wants to log in to the Employee Self-Service Portal.

Consider the following scenarios.

Example 1:
1. A user is not logged in, and then tries to access a record using a specific SYS ID in the URL.
2. The system redirects the user to the login page.
3. Rather than logging in, the user tries to access another site, such as the Employee Self-Service (/ess) Portal.
4. The system redirects the user to the login page again.
5. The user logs in and is redirected to the record that the user was first trying to access rather than the Employee Self-Service Portal.

Example 2:
1. A user is not logged in, and then tries to access a record using a specific SYS ID in the URL through the Employee Self-Service (/ess) Portal.
2. The system redirects the user to the login page.
3. Rather than logging in, the user tries to access another record through the Employee Self-Service Portal.
4. The system redirects the user to the login page again.
5. The user logs in and is redirected to the first record rather than the second.

Failed login attempts

Two inactive script actions are provided that allow a site to manage the number of times a user can fail to provide the correct password before getting locked out of the system.

From the left navigation pane, select System Policy > Script Actions to see/activate these script actions.

- SNC User Clear - updates the user record upon a successful login, resetting the number of failed attempts and updating the date of the last login
- SNC User Lockout Check - keeps track of the number of failed login attempts and will lock the user out after 5 failed attempts (change the number as desired)
Failed login attempts log

Attempts to log in are captured in event logs, and you can view the failed login attempts.

To view event logs, navigate to **System Policy** → **Event Logs**. You can filter for `login.failed` in the name field, and view the attempted login name, date, and IP address logged from the attempt.

![Failed login log](image)

**Figure 694: Failed login log**

Change settings for the Remember Me checkbox and cookie

When the **Remember me** check box is selected at login, a cookie is stored on the user's computer. This cookie automatically authenticates the user upon subsequent visits.

If the user logs out, the cookie is destroyed. The default value of the **Remember me** check box is controlled by one property, and whether or not the check box appears on the login page is controlled by a different property.

Change the default value of the Remember Me check box

You can change the default value of the **Remember Me** check box.

**Role required: admin**

1. Navigate to **System Properties** → **UI Properties**.
2. Locate the Default value of "Remember me" checkbox on login page property (glide.ui.remember.me.default).
3. To set the default value of the **Remember me** check box to No, clear the property check box.
4. To restore the default value of the **Remember me** check box to Yes, select the property check box.

Remove the Remember Me check box

You can remove the **Remember Me** check box so users do not have access to this feature.
Role required: security_admin

1. Click the lock icon (🔒) by the user name and activate elevated privileges for the security_admin role.


3. Locate the Remove "Remember me" checkbox from login page property (glide.ui.forgetme).

4. Select the property check box.
   This setting removes the Remember me check box, invalidates existing cookies, and disables Remember me functionality entirely.

5. To restore the Remember me check box to the login page, clear the property check box.

Self service password reset

The Self Service Password Reset plugin enables end users who are locally authenticated to reset their own passwords.

By default there is a new message and link on the login page to access the reset page. Users are prompted for their username and email address. If the system locates an active record, it generates a temporary password and sends an email to the user with the details. Users who reset their passwords receive a message with the results of the reset request.

Note: This feature works only for locally authenticated users. Users logging in via an SSO solution or an LDAP integration cannot reset their passwords in this fashion since the instance does not know these passwords.

The following functionality is updated with the installation of the Self Service Password Reset plugin:

- **Script include**
  PasswordResetAJAX - AJAX Script to process a password reset from a UI Page as guest. You can customize it to perform different user validation logic.

- **UI page**
  reset_password - Page to prompt for password reset information.

- **Event**
  password.reset - Event fired by PasswordResetAJAX script include to trigger the Password Reset Notification.

- **Email notification**
  Email sent to the user notifying the user of the reset and their temporary password.

- **Welcome page content**
  Message on the welcome page to link users to the password reset page.

- **Public page**
  Enabled the reset_password UI Page to be viewed by guests.

The components are fully customizable to meet individual needs:

- Password reset message in the login screen
- Contents of the Reset My Password form
- Password reset validation and processing
- Subject line and message in the email notification
Resetting a password

The Self Service Password Reset plugin creates a link on the login page that invites users who have forgotten their credentials to reset their password.

The user clicks the link on the login page to begin the password reset process.

- **Forgot your login credentials?**
- **If you cannot remember your login credentials, you can reset your password here.**

Figure 695: Password reset link on login page

The link from the login page opens the Reset My Password form.

- **Reset My Password**
- **Enter your User Name:**
- **Enter your Email Address:**

![Reset Password button](image)

- **Click here to return to the login page**

Figure 696: Password reset form

After the user completes the form and clicks **Reset Password**, the user_id and email address are validated. If they match, a temporary password is generated for that user and notification is sent to the email address.

Figure 697: Example of the out-of-box notification
Hello Steve,

A password reset was requested for your user account on the following Service-now.com system. [https://demosw.service-now.com/](https://demosw.service-now.com/)
The password has been reset and you can now log in with the following credentials:

User ID: sward
Password: KTpKvsvo

If you did not request this password reset, please let us know.

Best Regards,
Customer Support

Ref: MSG000033

The user logs in to the instance using the temporary password. The user is required to create a new, permanent password before continuing.

Figure 698: Change temporary password page

When the user clicks **Submit**, a session is established.

Add a password reset message

Add a custom message to a password reset screen to give users additional information or instructions.
Role required: admin

For example, security policies change and everyone is flagged for a password reset so they must create a new password following stronger rules. A user attempts to login, receives a message stating their
password must be changed, and is taken to a new page. You can add a custom message on the new
page.

1. Navigate to System UI > Messages.
2. For the Invalid Password, find the key login_invalid and update the message field.
3. For the Enter New Password, find the key System administrator requires you to change your
password and update the message field. (If that key is not available, create the key with Key =
System administrator requires you to change your password.)

ServiceNow access control

This SNC Access Control plugin (com.snc.snc_access_control) enables customers to control which
ServiceNow employees may access their instance, and when.

When the plugin is first activated, ServiceNow employees cannot log into the customer's instance. Any
currently logged-in ServiceNow employees remain logged in. The customer creates records in the SNC
Access Control table that grant access to specific SNC employees or all employees.

Login security

Security for authorized ServiceNow employee logins to customer instances employs encrypted tokens
generated by a secure server. Only properly authenticated ServiceNow employees are granted access
to a customer instances. Without the SNC Access Control plugin, the security server ensures that access
rights are enforced on hi.service-now.com. When the plugin is enabled, the encrypted login tokens must
match names in the plugin-provided access list, using the criteria defined in those records. This method of
authentication enables our customers to determine precisely which ServiceNow employees may access
their instances, and when these employees may do so.

The architecture chosen for this system has several features designed to enhance security for our
customer's instances:

- **Security server:** The security server is a locked-down, Linux host that can only be accessed by
  ServiceNow security personnel. This server is the only system that has access to the critical private
  encryption key necessary to produce the login tokens. By using this compartmentalization (a standard
  security practice), the private key is protected, even in the unlikely event that the HI instance is
  compromised by an attacker.

- **Synthetic users:** The facility on customer instances that enables authorized ServiceNow employees to
  log into their instance does not require an account to be provisioned on that instance. There is no user
  record provisioned, and no permanent or persisted credentials. Instead, a synthetic user is created for
  each ServiceNow employee logon. This user exists only in memory and provides no ongoing privileges.
  ServiceNow can deauthorize any ServiceNow employee at any time, and if the SNC Access Control
  plugin is enabled, our customers can also deauthorize any ServiceNow employee at any time.

- **Tokens:** The security tokens are specific to a customer instance and a particular ServiceNow
  employee. In addition, the mechanism that generates the tokens only works with actual ServiceNow
  employee logins to HI, not impersonated users. Once a security token is generated, it may only be used
  by a specific ServiceNow employee to log into a particular customer instance.

- **Time limit:** Security tokens expire four hours after they are generated. This limits the utility of hijacked
  tokens, which can only be used during this short window.

- **Logging:** Logins by ServiceNow employees to customer instances are recorded as a login event,
  and every action taken by the logged-in ServiceNow employee is added to both the transaction log
  (in the database) and the instance log (on the file system, which is inaccessible to most ServiceNow
  employees). ServiceNow employee logins and actions are readily identifiable, since the user names
  all end in @snc (like frodo.baggins@snc). This provides our customers with easy-to-use, robust, and
  reliable security logging for non-employee access.
When a ServiceNow employee wants to log into a customer instance, the security processing flow is as follows:

1. A ServiceNow Support technician requests a login for the customer's instance through hi.service-now.com.
2. HI checks that the technician has the proper role authorizing access to customer instances.
3. If the user has the proper role, HI sends the request for access to the Security Server.
4. The Security Server verifies that the request came from HI's IP address, and evaluates the request (user, role, and IP address of the requestor). If the request is valid, the Security Server approves it and constructs a token. This token contains the user name, roles on the customer instance, the instance ID, and the time (the start of the 4 hour token life span). Finally, the Security Server encrypts the token with the private encryption key.
5. The Security Server sends the encrypted token to HI.
6. HI sends the token to the Support technician's browser.
7. The Support technician's browser initiates a login into the customers instance, using the special user name ending with @snc.
8. The customer's instance uses the public key to decrypt the token. To verify the token, the instance matches it to the user name supplied in the previous step, the instance ID, and the authorized time window. If the SNC Access Control plugin is enabled, the instance verifies that the user is:
   - Listed
   - Active
   - Configured to access the instance in the current time window
9. If the user is authenticated, the customer instance creates a synthetic user in memory with the given roles. This user does not persist after the time limit expires, the user logs off, or the instance is restarted.
Audit logging

The following logging tracks logins and activity by ServiceNow employees:

- **Event logs:** The event logs show all ServiceNow logins to a customer instance.
- **Transaction logs:** The transaction logs show all activity on the instance, including any efforts to delete logs.

Request ServiceNow access control

Customers must request the SNC Access Control plugin (com.snc.snc_access_control) from HI.

Role required: admin

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
</tbody>
</table>
Specify the date and time you would like this plugin to be enabled

Date and time must be at least 2 business days from the current time.

**Note:** Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.

| Reason/Comments | Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows. |

3. Click **Submit**.

**Configure ServiceNow access control**

Configure an access control record to specify one or more ServiceNow employees who have permission to log into your instance.

Role required: admin

1. Navigate to **System Security > SNC Access Control**.
2. Click **New**.
3. Fill in the form fields (see table).
4. Click **Submit**.

**Table 580: SNC Access Control form fields**

<table>
<thead>
<tr>
<th>Form fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Names one or more ServiceNow employees who have permission to log into this instance. The names must have user records in <strong>hi.servicenow.com</strong> and are expressed as first name/last name in lower case letters separated by a dot. If this field contains an asterisk (*) anywhere within it, then all ServiceNow employees with customer logon rights are given access rights by this record. Otherwise, this field is treated as a comma-separated list of ServiceNow user names. For example, if this field contains <strong>beth.anglin,don.goodliffe,gregg.guevarra</strong>, then this record grants login permission to Beth Anglin, Don Goodliffe, and Gregg Guevarra. This is a mandatory field.</td>
</tr>
</tbody>
</table>
Form fields | Description
--- | ---
Reason | Human-readable field that describes why permission is being granted. This field is optional.
Start | Specifies the start date and time of the period during which the specified ServiceNow employees have login access. This field is mandatory.
End | Specifies the ending date and time of the period during which the specified ServiceNow employees have login access. This field is mandatory.
Active | Controls whether this permission record is active. The default is active.

Audit logging
The following logging tracks logins and activity by ServiceNow employees.

| Event logs | The event logs show all ServiceNow logins to a customer instance. |
| Transaction logs | The transaction logs show all activity on the instance, including any efforts to delete logs. |

Access control rules
Access control rules, also known as access control lists (ACL) or access controls, restrict access to data by requiring users to pass a set of requirements before they can interact with it.

All access control rules specify:
- The object and operation being secured
- The permissions required to access the object

Objects
Each object consists of a type and name that uniquely identifies a particular table, field, or record. For example, all these entries specify an object.

Table 581: Sample access control objects

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Object secured</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>[incident],[-- None --]</td>
<td>The Incident table.</td>
</tr>
<tr>
<td>record</td>
<td>[incident],[active]</td>
<td>The Active field in the Incident table.</td>
</tr>
</tbody>
</table>
### Operations

Each operation describes a valid action the system can take on the specified object. Some objects such as records support multiple operations, while other objects such as a REST_Endpoint only support one operation. For example, all these entries specify an operation.

#### Table 582: Sample access control operations

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Operation</th>
<th>Operation secured</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>[incident].[-- None --]</td>
<td>create</td>
<td>Creating records in the Incident table.</td>
</tr>
<tr>
<td>record</td>
<td>[incident].[active]</td>
<td>write</td>
<td>Updating the Active field in the Incident table.</td>
</tr>
<tr>
<td>REST_Endpoint</td>
<td>user_role_inheritance</td>
<td>execute</td>
<td>Running the user_role_inheritance Scripted REST API.</td>
</tr>
</tbody>
</table>

### Permissions

The permissions specify when someone can access the named object and operation. Security administrators can specify permission requirements by adding:

- One or more user roles to the Requires role list.
- One or more conditions.
- A script that evaluates to true or false or sets the answer variable to true or false.

To gain access to an object and operation, a user must pass all permissions listed in an access control. For example, this access control restricts access to write operations on the incident table.
### Access Control Incident

<table>
<thead>
<tr>
<th>Type</th>
<th>Application</th>
<th>Active</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>write</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Definition**

Access Control Rules allow access to the specified resource if all three of these checks evaluate to true:

1. The user has one of the roles specified in the Role list, or the list is empty.
2. Conditions in the Condition field evaluate to true, or conditions are empty.
3. The script in the Script field (advanced) evaluates to true, or sets the variable “answer” to true, or is empty.

The three checks are evaluated independently in the order displayed above.

**More Info**

<table>
<thead>
<tr>
<th>Requires role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Condition**

30 records match condition

<table>
<thead>
<tr>
<th>Add Filter Condition</th>
<th>Add &quot;OR&quot; Clause</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident state</td>
<td>is not Closed</td>
</tr>
</tbody>
</table>

**Update**  **Delete**
To update a record in the incident table, a user must have the listed role and the record must meet the condition.

Table 583: Permissions required to write to the incident table

<table>
<thead>
<tr>
<th>Permission type</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires role</td>
<td>Requires role: itil</td>
<td>Only allow users with the itil role to update incidents.</td>
</tr>
<tr>
<td>Condition</td>
<td>[Incident state] [is not] [Closed]</td>
<td>Only allow updates to active incident records.</td>
</tr>
</tbody>
</table>

Granting or denying access

The system evaluates access controls using this process.

Figure 701: Process to grant or deny access
Whenever a session requests data, the system searches for access control rules that match the requested object and operation. If there is a matching access control rule, then the system evaluates if the user has the permissions required to access the object and operation. If an access control rule specifies more than one permission, then the user must meet all permissions to gain access to the object and operation. Failing any one permission check prevents the user from accessing the matching object and operation.

If a user does not meet the permissions of the first matching rule, the system evaluates the permissions of the next matching access control rule as specified by the access control processing order. If the user fails to meet the permissions of any matching access control rule, the system denies access to the requested object and operation.

Note: If there are no matching access control rules for the requested object and operation, then the system grants the user access to it. In practice, it is rare for the system to find no matching rules because the system has a set of default access control rules that protect all record operations.

Roles

Normal admin users can view and debug access control rules. However, to create or update existing access control rules, administrators must elevate privileges to the security_admin role.

Process order for record ACL rules

Record ACL rules are processed in a certain order.

Record ACL rules are processed in the following order:

• Match the object against field ACL rules.
• Match the object against table ACL rules.

This processing order ensures that users gain access to more specific objects before gaining access to less specific ones.

A user must pass both field and table ACL rules in order to access a record object.

• If a user fails a field ACL rule but passes a table ACL rule, the user is denied access to the field described by the field ACL rule.
• If a user fails a table ACL rule, the user is denied access to all fields in the table even if the user previously passed a field ACL rule.
Figure 702: ACL matching

Start

User requests access to a record object

Find ACL rules matching object

Find first matching FIELD ACL Rule

1. ACL name matches table.field?
2. ACL name matches parent table.field?
3. ACL name matches *.field?
4. ACL name matches table.*?
5. ACL name matches parent table.*?
6. ACL name matches *.*?

No matches

Match found

Evaluate ACL Permissions

Pass

Find first matching TABLE ACL Rule

1. ACL name matches table?
2. ACL name matches parent table?
3. ACL name matches *?

No matches

Match found

Evaluate ACL Permissions

Pass

Grant user access to the record object

Fail

Deny user access to the record object

End
Processor ACL rules

ACL rules can secure access to the execute operation of all or specific processors.

Processor ACL rules specify the processor you want to secure. Use the asterisk character as a wildcard to search for any processor. For a list of available processors, navigate to System Definition > Processors.

By default, an ACL rule for the EmailClientProcessor is included to restrict the email client to users with the itil role.

Field ACL rules

Field ACL rules are processed in a certain order.

Field ACL rules are processed in the following order:

1. Match the table and field name. For example, incident.number.
2. Match the parent table and field name. For example, task.number.
3. Match any table (wildcard) and field name. For example, *.number.
4. Match the table and any field (wildcard). For example, incident.*.
5. Match the parent table and any field (wildcard). For example, task.*.
6. Match any table (wildcard) and any field (wildcard). For example, *.*.

The first successful evaluation stops ACL rule processing at the field level. This means that when a user passes a field ACL rule, the system stops searching for matching field ACL rules. The user must also pass the table ACL rules to be granted access to the record object. For example, if a user passes the field ACL rule for incident.number, the system stops searching for rules that secure the Number field. The user must then pass the table ACL rules on incident to see the Number field.

Record ACL rules

Record ACL rules consist of table and field names.

- Table name: the table being secured. If other tables extend from this table, then the table is considered a parent table. ACL rules for parent tables apply to any table that extends the parent table.
- Field name: the field being secured. Some fields are part of multiple tables because of table extension. ACL rules for fields in a parent table apply to any table that extends the parent table.

ACL rules can secure the following record operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>execute</td>
<td>Allows users to run an application or script.</td>
</tr>
<tr>
<td>create</td>
<td>Allows users to insert new records (rows) into a table.</td>
</tr>
<tr>
<td>read</td>
<td>Allows users to display records from a table.</td>
</tr>
<tr>
<td>write</td>
<td>Allows users to update records in a table.</td>
</tr>
<tr>
<td>delete</td>
<td>Allows users to remove records from a table or drop a table.</td>
</tr>
<tr>
<td>edit_task_relations</td>
<td>Allows users to extend the Task table.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>edit_ci_relations</td>
<td>Allows users to extend the Configuration Item [cmdb_ci] table.</td>
</tr>
<tr>
<td>save_as_template</td>
<td>Allows users to save a record as a template.</td>
</tr>
<tr>
<td>add_to_list</td>
<td>Allows users to insert records (rows) into a table from a list.</td>
</tr>
<tr>
<td>list_edit</td>
<td>Allows users to update records (rows) from a list.</td>
</tr>
<tr>
<td>report_on</td>
<td>Allows users to create reports on tables. This operation is not valid for field ACL rules.</td>
</tr>
<tr>
<td>personalize_choices</td>
<td>Allows users to configure the table or field.</td>
</tr>
</tbody>
</table>

Table ACL rules

In most cases there is not an individual field ACL rule for every field in the table the users is trying to access.

If no field ACL rule matches the record object, the user must pass the table ACL rule. Since the base system includes wildcard table ACL rules that match every table, the user must always pass at least one table ACL rule. The base system provides additional table ACL rules to control access to specific tables.

Table ACL rules are processed in the following order:
1. Match the table name. For example, incident.
2. Match the parent table name. For example, task.
3. Match any table name (wildcard). For example, *.

Just like with field ACL rules, the system grants the user access to the record object secured by the ACL rule and stops searching for matching ACL rules the first time a user passes a table ACL rule's permissions. A user who passes the table ACL rule for incident has access to all fields in the Incident table. A user who passes the table ACL rule for task has access to all fields in the Task table as well as the fields in extended tables. A user who passes the table ACL rule for any table has access to all fields in all tables.

UI page ACL rules

UI page ACL rules specify the UI page to be secured.

For a list of available UI pages, navigate to System UI > UI Pages. When defining an ACL rule for a UI page, use the fully scoped page name. For example, x_myapp_mypage.

**Note:** Starting with the Geneva release, you can use the wildcard * character in the Name field on ui_page type ACLs to match any UI pages. Prior to Geneva, the wildcard character was not supported for UI page ACLs.

ACL rules can secure the following UI page operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>execute</td>
<td>Allows users to run an application or script.</td>
</tr>
<tr>
<td>create</td>
<td>Allows users to insert new UI page records.</td>
</tr>
<tr>
<td>read</td>
<td>Allows users to display the UI page.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>write</td>
<td>Allows users to update UI page records.</td>
</tr>
<tr>
<td>delete</td>
<td>Allows users to remove UI page records.</td>
</tr>
<tr>
<td>edit_task_relations</td>
<td>Allows users to extend the Task table.</td>
</tr>
<tr>
<td>edit_ci_relations</td>
<td>Allows users to extend the Configuration Item [cmdb_ci] table.</td>
</tr>
<tr>
<td>save_as_template</td>
<td>Allows users to save a UI page record as a template.</td>
</tr>
<tr>
<td>add_to_list</td>
<td>Allows users to insert UI page records from a list.</td>
</tr>
<tr>
<td>list_edit</td>
<td>Allows users to update UI page records from a list.</td>
</tr>
<tr>
<td>report_on</td>
<td>Allows users to create reports on UI page records.</td>
</tr>
<tr>
<td>personalize_choices</td>
<td>Allows users to configure UI page records.</td>
</tr>
</tbody>
</table>

Since UI pages typically only display read-only information, the most common UI page ACL rule is for the "read" operation.

**ACL rules in scoped applications**

You can create ACL rules for objects in the same scope as the ACL rule and for tables with at least one field that is in the same scope as the ACL rule.

You can create ACL rules for objects in the same scope as the ACL rule and for tables with at least one field that is in the same scope as the ACL rule. For tables that are in a different scope than the ACL rule record, the types of rules are limited.

- You can create an ACL rule for any table, UI page, or other object that is in the same scope as the ACL rule.
- You can create an ACL for a field that is in the same scope as the ACL rule.
  - If the table is in the same scope, you can use a script to evaluate permissions.
  - If the table is in a different scope, you cannot use a script to evaluate permissions.
- You cannot create or modify ACL rules for objects that are in a different scope than the application you have selected in the application picker, including adding a role to an ACL in a different scope.
- You can create wildcard table rules (*) only in the global scope.
- You can create wildcard field rules (*) only for tables in the same scope as the ACL rule.

**Client-callable script include ACL rules**

ACL rules can secure access to the execute operation of all or specific client-callable scripts.

Script include ACL rules specify the client-callable script include to be secured. Use the asterisk character as a wildcard to search for any client-callable script include. For a list of available script includes, navigate to System Definition > Script Includes. You can personalize the list to show the Client callable column.

The base system does not include any ACL rules for client-callable script includes.
Evaluate the admin override at the access level

If you want to force ACL evaluation for admin overrides at the access level, you can add a system property. Role required: security_admin

ACLs are evaluated cumulatively. If there are number of ACLs on any given field and the Admin Overrides option is false (not selected) on one of them, then the effective admin overrides for all the ACLs are considered to be false. This causes admins to be unable to pass even the ACL where the override should be in effect.

- Add the following property to the system properties table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
</table>
| glide.security.admin.override.accessterm | Evaluates the admin override condition at the access term level. | Type: true | false
| Default value: true for new instances, false for upgrades. Location: Add to the System Properties [sys_properties] table |

Contextual security

The contextual security manager provides incredible flexibility and power to protect information by controlling read/write/create/delete authorization.

Key advantages

- **Contextual security**: Secure a record based on its contents
- **Hierarchical security**: Can apply security rules to any level in our object hierarchy

Everything you can do with the simple security manager you can also do with the contextual security manager. Likewise, after conversion to the contextual security manager, you should not see any behavior changes in your instance.

Securing Fields and Tables

Under the simple security manager, you could secure fields and tables by adding roles to the appropriate dictionary entry. After installing the contextual security manager, these dictionary roles are no longer tested. Instead the system looks for ACL rules on fields and or tables.

**Warning**: After you install the Contextual Security Manager you must secure fields and tables via ACL rules. Even if you configure the dictionary form and add roles to a dictionary entry, no change in rights will occur.
Granting Roles to Users

Roles can still be granted to users or groups using the same logic as under the simple security manager. The one noteworthy exception is that the "roles" field on the user record is no longer checked under the contextual security manager (and should be, in fact, removed from your user and group forms upon installation).

**Note:** To add roles to a user or group record under Contextual Security you must add them to the Roles related list instead of to the user or group record itself.

Applications and Modules

Applications and modules both contain lists of roles under which they can be viewed. For example, the System Definition application requires the admin role to be viewed.

Security rights for Applications and Modules are still defined via these role arrays although they may be transitioned to ACLs at some future date.

Catalog Items and Variables

Both catalog items and catalog variables contain lists of roles under which they can be viewed.

Security rights for these entities are still defined via these role arrays although they may be transitioned to ACLs at some future date.

Inheritability of Group Roles

Under the contextual security manager, a group still automatically inherits any role granted to the group.

**Note:** The role's inherits flag is set to true.

Plugins

These plugins are automatically installed on new instances and can be activated for upgrades:

- **Contextual Security:** Provides contextual security functionality.
- **Contextual Security: Role Management Enhancements:** Eliminates duplicate inherited entries in the User Roles [sys_user_has_role] table (starting with the Geneva release) by tracking the number of times a role is inherited from another role or group. Once activated, inherited entries in the User Roles table are automatically managed and cannot be deleted directly. Instead, a containing role or group is expected to be removed from the user.

Contextual Security: Role Management Enhancements

Contextual Security: Role Management Enhancements prevents duplicate entries in the User Roles [sys_user_has_role] table.

Contextual Security: Role Management Enhancements is automatically installed on new instances starting with the Geneva release and can be activated for upgrades. When activated, Contextual Security: Role Management Enhancements replaces the legacy version.
Eliminate duplicate entries through inheritance count

The Contextual Security: Role Management Enhancements uses the inheritance count (inh_count) column to track the number of times a role is inherited from another role or group. In the User Roles [sys_user_has_role] table, a user can inherit a specific role only one time, eliminating duplicate entries. The inheritance count (inh_count) column is read-only and is calculated by the number of times the role is inherited by the user. If the inh_count value is 0, the inherited role is removed from the user.

When Contextual Security: Role Management Enhancements is activated, the following columns are deprecated, but remain in the User Roles table for backward compatibility:

- granted_by
- included_in_role
- included_in_role_instance

Visualize role inheritance through the Role Inheritance Map

The Role Inheritance Map displays a visual representation of inherited roles. To view the Role Inheritance Map, configure the User Roles [sys_user_has_role] table to display the Role Inheritance Map column.

Figure 703: Role Inheritance Map

Upgrade to Contextual Security: Role Management Enhancements
Contextual Security: Role Management Enhancements is automatically installed on new instances starting with the Geneva release. You can upgrade from the legacy Contextual Security plugin to Contextual Security: Role Management Enhancements.

Role required: admin

The contextual security plugins are active on new instances.
### Table 587: Plugins for Contextual Security: Role Management Enhancements

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual Security: Role Management Enhancements</td>
<td>Prevents duplicate entries in the User Roles [sys_user_has_role] table.</td>
</tr>
<tr>
<td>[com.glide.role_management.inh_count]</td>
<td></td>
</tr>
<tr>
<td>Contextual Security: Role Management Enhancements REST API</td>
<td>Enables API functionality for role management. If not already active, this plugin activates Contextual Security: Role Management Enhancements.</td>
</tr>
<tr>
<td>[com.glide.role_management.inh_count.rest_api]</td>
<td></td>
</tr>
</tbody>
</table>

1. Navigate to **System Definition > Plugins**.
2. Find and click the plugin name.
3. On the System Plugin form, review the plugin details and then click the **Activate/Upgrade** related link.
4. Click **Activate**.

*Enable role auditing with Contextual Security: Role Management Enhancements*

Set a system property to allow the Audit Roles table to create audit records related to user roles.

Role required: admin

The Audit Roles [sys_audit_role] table automatically maintains changes to user records. If the Contextual Security: Role Management Enhancements [com.glide.role_management.inh_count] plugin is installed, you must set a system property to true to enable role auditing.

2. Set the following property to true.

<table>
<thead>
<tr>
<th>System property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.role_management.v2.audit_roles</td>
<td>If the Contextual Security: Role Management Enhancements [com.glide.role_management.inh_count] plugin is installed, setting this property to true allows the Audit Roles [sys_audit_role] table to create records when user roles change. This property is available in Geneva Patch 9 Hot Fix 3.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Options</strong>: true</td>
</tr>
<tr>
<td></td>
<td>- <strong>Default</strong>: false</td>
</tr>
<tr>
<td></td>
<td>- <strong>Learn more</strong>: <a href="#">Audit user roles</a> on page 1744</td>
</tr>
</tbody>
</table>

### ACL configuration watcher

The ACL configuration watcher lets you know what related ACLs exist on a table when you insert, update, or delete an ACL on the same table.

The ACL configuration watcher is an interceptor window that displays every time you make important changes on the Access Control [sys_security_acl] table. It displays a security rules summary window.
where you can view ACLs related to the one you are modifying. You can not modify any ACLs from the security rules window. To make any modifications, close the watcher window and go to those ACLs.

The ACL configuration watcher is available with the Geneva release.

The ACL configuration watcher does not appear in the following situations:

• If you save or update an ACL record without actually making any changes.
• If you make minor updates (not an insert or delete), such as updating scripts, conditions, and the admin-overrides option.
• If the ACL record is not active.

**ACL Security Rules window**

The configuration watcher shows the *ACL execution plan*. The execution plan is displayed in the security rules pop-up window. You can view this kind of information:

Table 588: ACL configuration window elements

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>red highlight</td>
<td>An ACL that is deleted or deactivated.</td>
</tr>
<tr>
<td>blue highlight</td>
<td>An ACL that is modified.</td>
</tr>
<tr>
<td>green highlight</td>
<td>An ACL that is added or becomes active.</td>
</tr>
<tr>
<td>Masked</td>
<td>An ACL that was effective until you made a change.</td>
</tr>
<tr>
<td>Unmasked</td>
<td>An ACL that was just made effective when you made a change.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
Show ACL execution plan

Administrators can view how ACLs relate to each other by viewing an execution plan for any ACL in the instance.

Role required: security_admin

1.  *Elevate to a privileged role* on page 2447.
2.  Open an ACL that is a record-type ACL.
3.  Click *Show ACL Execution Plan*.

   The security rules window appears for the ACL.
Security Rules for "vtb_task.description"

Table 589: ACL execution plan window

<table>
<thead>
<tr>
<th>UI item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>The name of the ACL.</td>
</tr>
<tr>
<td>Tab name</td>
<td>If the ACL is create, read, write, or delete.</td>
</tr>
<tr>
<td>Row level</td>
<td>Row-level ACLs that run on this table.</td>
</tr>
<tr>
<td>Field level</td>
<td>Field-level ACLs that run only on this field (or column in the table).</td>
</tr>
</tbody>
</table>

4. Click **Show all** to show all related ACLs, including those that are overridden and generic ACLs that apply to all records. Overridden ACLs have a line through the name and generic ACLs have the wildcard character asterisk (*) for the name.

5. Click **Show Effective** to show only the immediate ACLs related to the one you are viewing, not the ACLs that are on tables from which the ACL table is extended, nor the generic wildcard (*) ACLs.

Use the ACL configuration watcher

Use the ACL configuration watcher after you elevate to security administrator.

Role required: security_admin

_Elevate to a privileged role_ on page 2447

1. Open an ACL that is a record-type ACL.
2. Perform an action on the ACL, such as modifying it, or selecting an option from the context menu like **Insert**.

3. If you modified any values on the Access Control form, right-click the header and select **Save** or click **Update** or **Delete**.

The Security Rules window appears. The system did not yet perform the database action on the ACL, so the changes are not yet saved.

These are examples of security rules on the Visual Task Board application's Private Task [vtb_task] table. See **ACL configuration watcher** on page 2510 for a description of the items on this window.

---

### Verify Security Rules for "vtb_task.short_description"

<table>
<thead>
<tr>
<th>Row level</th>
<th>Field level</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>vtb_task</td>
<td>vtb_task.short_description</td>
<td>Deactivated</td>
</tr>
<tr>
<td></td>
<td>task.short_description</td>
<td>Unmasked</td>
</tr>
</tbody>
</table>

---

### Verify Security Rules for "vtb_task"

<table>
<thead>
<tr>
<th>Row level</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>vtb_task</td>
<td>Added</td>
</tr>
</tbody>
</table>

---

### Verify Security Rules for "vtb_task"

<table>
<thead>
<tr>
<th>Row level</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deleted</td>
</tr>
<tr>
<td>vtb_task</td>
<td>Unmasked</td>
</tr>
<tr>
<td>*</td>
<td>Unmasked</td>
</tr>
<tr>
<td>*</td>
<td>Unmasked</td>
</tr>
</tbody>
</table>
4. Just as with the execution plan, you can click **Show all** to show all related ACLs, including those that are overridden and generic ACLs that apply to all records, or click **Show Effective** to show only the immediate ACLs related to the one you are viewing.

5. Hover your mouse over any of the ACLs to see a description.

---

Create an ACL rule

Create custom ACL rules to secure access to new objects or to change the default security behavior.

**Role required:** security_admin

To create new ACL rules, you must elevate privileges to the security_admin role. For tables that are in a different scope than the ACL rule record, the types of rules are limited.

1. Elevate privileges to the security_admin role.
2. Navigate to **System Security > Access Control (ACL).**
3. Click **New.**
4. Define the object the ACL rule secures and the permissions required to access the object.
5. Right-click the form header and select **Save.**
Access Control Rules allow access to the specified resource if all three of these checks evaluate to true:

1. The user has one of the roles specified in the Role list, or the list is empty.
2. Conditions in the Condition field (advanced) evaluate to true, or return the variable "answer" to true, or is empty.
3. The three checks are evaluated independently in the order displayed above.

### Definition

<table>
<thead>
<tr>
<th>Type</th>
<th>Operation</th>
<th>Action Status</th>
<th>Active</th>
<th>Admin overrides</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>read</td>
<td>Global</td>
<td></td>
<td></td>
<td>Incident</td>
<td>Information required to read incident records</td>
</tr>
</tbody>
</table>

### Conditions

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>- choose field -</td>
<td>- open -</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Select what kind of object this ACL rule secures. The type of object determines how the object is named and what operations are available. This field becomes read only after the ACL rule is created (starting with the Geneva release). If you want to change the type, you must delete the ACL and create a new one with the correct type.</td>
</tr>
<tr>
<td><strong>Operation</strong></td>
<td>Select the operation this ACL rule secures. Each object type has its own list of operations. An ACL rule can only secure one operation. To secure multiple operations, create a separate ACL rule for each. For a list of operations and descriptions, see <a href="#">Record ACL rules</a> on page 2504.</td>
</tr>
<tr>
<td><strong>Admin Overrides</strong></td>
<td>Select this check box to have users with the admin role automatically pass the permissions check for this ACL rule. Admin users pass regardless of what script or role restrictions apply. However, the <strong>nobody</strong> role takes precedence over the admin override option. If an ACL is assigned the <strong>nobody</strong> role, admin users cannot access the resource even when <strong>Admin overrides</strong> is selected. See <a href="#">Base system roles</a> on page 1724.</td>
</tr>
<tr>
<td><strong>Active</strong></td>
<td>Select this check box to enforce this ACL rule.</td>
</tr>
<tr>
<td><strong>Advanced</strong></td>
<td>Select this check box to display the <strong>Script</strong> field.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Name       | Enter the name of the object being secured, either the record name or the table and field names. The more specific the name is, the more specific the ACL rule is. You can use the wildcard character asterisk (*) in place of a record name, table name, or field name to select all objects that match a particular record type, all tables, or all fields. You cannot combine a wildcard character and a text search. For example, inc* is not a valid ACL rule name, but incident.* and *.number are valid ACL rule names.  
**Note:** Click the blue triangle to manually enter the record name or the table and field names of the object being secured. Use this option to secure an object that does not appear in the dropdown. |
| Description| [Optional] Enter a description of the object or permissions this ACL rule secures.                                                           |
| Requires role | Use this list to specify the roles a user must have in order to access the object. If you list multiple roles, a user with any one of the listed roles can access the object. The Requires role list appears as a related list.  
**Note:** Users with the admin role will always pass this permissions check because the admin role automatically grants users all other roles. |
| Condition  | Use this condition builder to select the fields and values that must be true for users to access the object. xl |
Enter a custom script describing the permissions required to access the object. The script can use the values of the current and previous global variables as well as system properties. The script must generate a true or false response in one of two ways:

- return an answer variable set to a value of true or false
- evaluate to true or false

In either case, users only gain access to the object when the script evaluates to true and the user meets any conditions the ACL rule has. Both the conditions and the script must evaluate to true for a user to access the object.

**Note:** If the item you are evaluating the ACL for is in a related list, current points to the item the related list is on instead of the current item the ACL is for. However, if the item you are evaluating the ACL for is not in a related list, current points to the actual item.

---

Grant or deny access

When a user attempts to access a particular object, the system searches for ACL rules that match the requested object's type, operation, and name.

If an ACL rule matches the requested object type, operation, or name, then the user must meet the permissions described in this rule to access the secured object.

If the user fails to meet the permissions required by the first rule, the system searches for the next matching ACL rule. For each matching ACL rule, the user has a chance to meet the required permissions in order to access the object. The system stops searching for matching ACL rules if the user ever meets a matching ACL rule's permissions. If the user cannot meet the permissions of any matching ACL rules, the system denies the user access to the object.

The effects of being denied access to an object depend on the ACL rule that the user failed. For example, failing a read operation ACL rule prevents the user from seeing the object. Depending on the object secured, the ACL rule could hide a field on a form, hide rows from a list, or prevent a user from accessing a particular UI page. See the table for a complete list of results of failing an ACL rule for a given operation and object type.

**Table 591: ACL rule table**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Results of Failing an ACL Rule on Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>execute</td>
<td>User cannot execute scripts on record or UI page.</td>
</tr>
<tr>
<td>Operation</td>
<td>Results of Failing an ACL Rule on Object</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>create</td>
<td>User cannot see the New UI action from forms. The user also cannot insert records into a table using API protocols such as web services. Note that a create ACL with a condition that a field contain a specific value always evaluates as false, as fields on new records are considered empty until saved.</td>
</tr>
<tr>
<td>read</td>
<td>User cannot see the object in forms or lists. The user also cannot retrieve records using API protocols such as web services.</td>
</tr>
<tr>
<td>write</td>
<td>User sees a read-only field in forms and lists, and the user cannot update records using API protocols such as web services.</td>
</tr>
<tr>
<td>delete</td>
<td>User cannot see the Delete UI action from forms. The user also cannot remove records from a table using API protocols such as web services.</td>
</tr>
<tr>
<td>edit_task_relations</td>
<td>User cannot define relationships between task tables.</td>
</tr>
<tr>
<td>edit_ci_relations</td>
<td>User cannot define relationships between Configuration Item [cmdb_ci] tables.</td>
</tr>
<tr>
<td>save_as_template</td>
<td>Used to control the fields that should be saved when a template is created.</td>
</tr>
<tr>
<td>add_to_list</td>
<td>User cannot view or personalize specific columns in the list mechanic.</td>
</tr>
<tr>
<td>list_edit</td>
<td>User cannot update records (rows) from a list.</td>
</tr>
<tr>
<td>report_on</td>
<td>User cannot create reports on the object.</td>
</tr>
<tr>
<td>personalize_choices</td>
<td>User cannot right-click a choice list field and select <em>Configure Choices</em>.</td>
</tr>
</tbody>
</table>

**Match ACL rules to objects**

Each object type has its own matching requirements.
### Table 592: Object type

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Matching ACL Rules Required to Access Object</th>
<th>Existing Wildcard ACL Rules</th>
</tr>
</thead>
</table>
| Client-callable script includes | Users must meet the permissions of two ACL rules:  
1. All wildcard ACL rules for the object (if any ACL rule exists for the operation).  
2. The first ACL rule that matches the object's name (if any ACL rule exists for the operation). | By default, there are no wildcard (*) rules for these object types. If you create a wildcard ACL rule for one of these objects, then the ACL rule applies to all objects of this type. |
| Processors                |                                                                                                                 |                                                                                                |
| UI pages                  | Users must meet the permissions of two ACL rules:  
1. The first ACL rule that matches the record's field (if any ACL rule exists for the operation).  
2. The first ACL rule that matches the record's table (if any ACL rule exists for the operation). | By default, there are wildcard table rules (*) for the create, read, write, and delete operations and wildcard field rules (*.*) for the personalize_choices, create, and save_as_template operations. When you create a new table, create new ACL rules for the table unless you want to use the provided wildcard ACL rules. |
| Record                    |                                                                                                                 |                                                                                                |

**Note:** The high security property Security manager default behavior (glide.sm.default_mode) determines whether users can access objects that only match against wildcard table ACL rules. When this property is set to Deny access, only administrators can access objects that match the wildcard table ACL rules.

**Note:** The wildcard field ACL rule (.*') for the create operation reuses the same permissions as the write operation. This means that the create permissions are the same as the write permissions unless you define an explicit create operation ACL rule.

Evaluate ACL rule permission requirements

An ACL rule only grants a user access to an object if the user meets all of the permissions required by the matching ACL rule.

- The condition must evaluate to true.
- The script must evaluate to true or return an answer variable with the value of true.
- The user must have one of the roles in the required roles list. If the list is empty, this condition evaluates to true.
- [Record ACL rules only] The matching table-level and field-level ACL rules must both evaluate to true.
Figure 705: ACL evaluate permissions
Control whether script conditions apply to reference fields

If you want to enable script conditions for reference fields, you can add a system property. The default behavior is intended to improve instance performance. If you want to enable script conditions for reference fields, add the following system property.

Table 593: System property

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.sys_reference_row_check</td>
<td>Controls whether the script conditions of Access Control Rules apply to a table’s reference fields.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: [sys_properties] table</td>
</tr>
</tbody>
</table>

Multiple ACL rules at the same point in the processing order

If two or more rules match at the same point in the processing order, the user must pass any one of the ACL rules permissions to access the object.

For example, if you create two field ACL rules for incident.number, then a user who passes one rule has access to the number field regardless of whether the user failed any other field ACL rule at the same point in the processing order.

ACL rule output messages

ACL debugging displays ACL rule output messages at the bottom of each list and form. The output message does the following:

• Improve readability
• Include context information
• Show the results of each type of ACL test
• To provide hyperlinks to the ACLs that run on the list or form.

Each message displays the following information:

Table 594: Message information

<table>
<thead>
<tr>
<th>Message element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>The total time used to process this ACL rule.</td>
</tr>
<tr>
<td>PATH</td>
<td>Information that uniquely identifies each ACL rule in the format: &lt;ACL rule type&gt;/&lt;ACL rule name&gt;/&lt;Operation&gt;.</td>
</tr>
<tr>
<td>CONTEXT</td>
<td>The object being evaluated by the ACL rule.</td>
</tr>
<tr>
<td>RC</td>
<td>The return code of the ACL rule. A true value passes the ACL rule. A false value fails the ACL rule.</td>
</tr>
<tr>
<td>Message element</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| RULE            | A brief summary of processors and scripts, followed by ACL results for each table-level and field-level ACL evaluation. Most ACL evaluations show an overall pass or fail result followed by a breakdown of the results for each type of ACL criteria:  
  • Role  
  • Condition  
  • Script |

The icons that appear show how the ACL was evaluated:

**Table 595: Evaluation icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A green checkmark (✅ #)</td>
<td>Indicates the table or field passed the criteria.</td>
</tr>
<tr>
<td>A red x icon (❌)</td>
<td>Indicates the table or field did not pass.</td>
</tr>
<tr>
<td>An empty gray circle icon (⊙ #)</td>
<td>Indicates the ACL evaluation did not need to be performed.</td>
</tr>
<tr>
<td>A blue checkmark, x, or empty circle</td>
<td>Indicates that the ACL was taken from a cached result of a previous ACL check. The icons mean the same as the above.</td>
</tr>
</tbody>
</table>

Click the name of the ACL next to any of the output messages to open that ACL record.
Access control rules debug

ACL rule debugging tools are available.

The following ACL rule debugging tools are available:

- Field level debugging
- ACL rule output messages

To enable ACL rule debugging, navigate to **System Security > Debug Security Rules**.

**Note:** Impersonation can simplify debugging ACL rules. First enable ACL debugging, then impersonate another user to see what ACL rules the user passes and fails.

Field level debugging

When debugging is enabled, a small bug icon appears beside each field with an ACL rule.

When debugging is enabled, a small bug icon (_bug_#) appears beside each field with an ACL rule. Clicking the icon lists the ACL rules that apply for the field and the evaluation results.
Impersonate users to debug ACLs

Impersonation can simplify debugging ACL rules.

After enabling ACL debugging, you can impersonate another user to see what ACL rules the user passes and fails. When you impersonate a user, you can only see what that user is allowed to see. For example, you cannot view a record that an ACL prevents the user from seeing.

To make debugging easier, read-only access to certain ACL-related tables is enabled by default, even when impersonating a user that does not have read access to the tables. To change this functionality, set the following property to false.

<table>
<thead>
<tr>
<th>System property</th>
<th>Description</th>
<th>Default setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.security.access_acl_as_impersonator</td>
<td>Allows read access to the following tables while impersonating a user: sys_security_acl, sys_security_operation, sys_security_type, and sys_user_role. As a result, the impersonating user can read data that the impersonated user cannot read.</td>
<td>true</td>
</tr>
</tbody>
</table>

**Note:** When the property is set to false, the impersonated user might be prevented from reading ACL-related data. In this case, a second session logged in as admin or security_admin might be required to debug ACLs.

ACL troubleshooting reference

A list of common ACL rule errors and their solutions.

Enable debugging to help troubleshoot an issue.
Table 596: Troubleshoot

<table>
<thead>
<tr>
<th>Error or symptom</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>You cannot access records from a custom table.</td>
<td>Create a table ACL rule for the custom table granting users access to the table. Without an explicit table ACL rule, users must pass the permissions in the table wildcard (*) ACL rule, which by default restricts access to administrators only. Enable debugging and determine what ACL rules are evaluated for the custom table.</td>
</tr>
<tr>
<td>You create a custom ACL rule that does not work properly.</td>
<td>The most likely problems are that another rule takes precedence over your custom rule in the processing order or that the user does not meet all the permission requirements for the object type. Enable debugging and verify that the ACL rule is being evaluated.</td>
</tr>
<tr>
<td>Your field ACL rule does not work properly.</td>
<td>There is likely a table ACL rule that the user has not met. Enable debugging and determine what ACL rules are evaluated for the field. Verify that there is not a conflicting table ACL rule or duplicate field ACL rule.</td>
</tr>
<tr>
<td>Your table ACL rule does not work properly.</td>
<td>There is either an ACL rule higher in the processing order or a duplicate table ACL rule interfering with the table ACL rule. Enable debugging and determine what ACL rules are evaluated for the table.</td>
</tr>
<tr>
<td>You can see a field in a list but not in form.</td>
<td>It is possible that the ACL rule conditions or script are being triggered in the list but not in the form. Enable debugging and determine when the ACL rules evaluate to true. Update the conditions or script to have the same behavior on the list and form.</td>
</tr>
<tr>
<td>You receive an error message when trying to execute a processor or client-callable script include.</td>
<td>There is an ACL rule for the processor or client-callable script include that the user has not met. If the user should have access to the object, enable debugging and determine what ACL rules are evaluated for the processor or script include. Update the ACL rule or the user roles as needed to access the object.</td>
</tr>
</tbody>
</table>

Mandatory roles

You can give both internal users and external users access to your instance. However, you might not want both types of users to have the same level of access. To provide added security, every user must have at least one role so the instance can distinguish between the users that are internal and the users that are external.

Prior to the Geneva release, ESS users had no role, but were still considered part of your organization and could access basic system resources by default, such as an ESS home page. Starting with the Geneva
release, ESS users can obtain the **snc_internal** role and still retain the same level of access they had prior to Geneva.

External users must obtain, at minimum, the **snc_external** role. This role indicates that the user is external to your organization and should not have any access to resources unless you explicitly allow access through ACLs for the **snc_external** role, or you explicitly grant them additional roles. By default, users with the **snc_external** role are unable to access non-record type resources as well, such as processors and UI pages.

You should not mark the **snc_internal** role as elevated. Otherwise, internal users could not access the instance.

---

**Note:** You can use encryption contexts with the **snc_internal** and **snc_external** roles.

---

**The Explicit Roles plugin**

The Explicit Roles (**com.glide.explicit_roles**) plugin provides the **snc_external** and **snc_internal** roles. This plugin is activated automatically when you activate the Customer Service Portal.

When this plugin is activated:

- All existing users are automatically assigned the **snc_internal** role. This role does not change existing access levels or system behavior. Rather, it provides a category to differentiate internal users from external users. All internal users maintain the same level of access as before the plugin was activated.
- All existing ACLs that do not have a role requirement are automatically assigned the **snc_internal** role. Because both existing ACLs and roles are assigned the **snc_internal** role, existing access levels do not change.
- External users must obtain, at minimum, the **snc_external** role to access the instance. This role is automatically assigned to external Customer Service Portal contacts. If the Customer Service Portal is not activated, this role must be manually granted to external users. Access to records is granted through ACLs.

**Providing access to external users**

You can grant external users access to tables be creating a set of ACLs for the table. See **Provide external users access to a table** on page 2529.

Another approach you can take is to give all external users access to all tables, and then restrict access to specific tables. You can do this by adding the **snc_internal** role to the *ACL that is of Type ui_page.

**The hasRoles() method**

The **hasRoles()** method is still available, but is deprecated in the Geneva release. Use the **hasRole(role name)** method instead.

If you use the **hasRoles()** method, note these changes:

- This method automatically excludes the default **snc_internal** role when it checks for roles. This means that if a user has only the **snc_internal** role, the **hasRoles()** method still returns **false**.
- If the user has the **snc_external** role, **false** is returned because the instance considers external users as without a role.
Activate the Explicit Roles plugin

Request activation of the Explicit Roles plugin on HI.
Role required: none
Request the plugin through the HI Service Portal.
1. In the HI Service Portal, click Service Catalog > Activate Plugin.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
</tbody>
</table>
| Specify the date and time you would like this plugin to be enabled | Date and time must be at least 2 business days from the current time. **Note:** Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.
| Reason/Comments | Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows. |

3. Click Submit.

Provide external users access to a table

To enable users with only the snc_external role to access the list view of a table, you must create a series of ACLs.
Role required: security_admin
1. Elevate to the security_admin role.
2. Create a new ACL with the following settings:
   - **Type:** ui_page
   - **Operation:** read
   - **Name:** `{table_name}_list`
   - **Required role:** snc_external
3. On the default read ACL for the table, add snc_external in the Required role list. Create the ACL if it does not already exist.
4. Use these settings to create another ACL:
   - **Type:** ui_page
   - **Operation:** read
   - **Name:** `{table_name}`
   - **Required role:** snc_external
5. Use these settings to create another ACL to give the user write access to a field in the table:

- **Type**: record
- **Operation**: create
- **Name**: `{table_name} {column_name}
- **Required role**: snc_external

Repeat this step for every field that you want to give the user write access to. Use an asterisk * instead of the column name to provide access to all fields at once.

### Access control analytic data

Usage Analytics measures how often access control rules allow or deny users to see database query results.

The system evaluates access control rules that use script-based permissions after fetching records from the database. If a user passes the script-based permissions, the system allows the user to see the records. If a user fails the script-based permissions, the system denies the user access by discarding any records from the results that match the access control rule.

To help determine how often the system is performing such unneeded database queries, Usage Analytics collects information about each row-level access control transaction:

- The table name.
- The results of the permission check as the value **allowed** or **denied**.
- The number of times row record access was allowed or denied.

Administrators can enable or disable the system collecting this data by setting a Usage Analytics **property**.

### ACL information sent to Usage Analytics

A property is available to control the sending of information to Usage Analytics.

When a user tries to access records that are restricted by ACLs, the discarded records can be sent to Usage Analytics. Administrators can add and enable a property to control the sending of this information to Usage Analytics. The property must be added to the System Properties [sys_properties] table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.security.event.acl_db_eval.send</td>
<td>When enabled, this property sends lists of restricted records to Usage Analytics.</td>
<td>false</td>
</tr>
</tbody>
</table>

### Encryption support

Encryption is a process that scrambles information into a format that unauthorized parties cannot decode or use.

Users who have access to the encryption context can see data encrypted with that particular encryption context. The encryption process requires an administrator to grant an encryption context to users by granting the user an associated role.
**Note:** Impersonation does not change the encryption contexts available to a user. Even while impersonating, you have only the encryption contexts available to you originally.

After encryption, text fields and attachments are no longer accessible by database tools and cannot be indexed. In addition, users cannot add encrypted fields to a filter. You can encrypt all String fields, including fields provided by default in the system and new fields that you create in the dictionary.

Users with the admin role can activate the Encryption Support plugin.

**Encryption context setup**

Administrators can create an encryption context that uses an encryption key.

Your instance can generate an encryption key, or you can generate your own key with a certificate authority. See your certificate authority documentation for information on creating an encryption key.

1. Navigate to **System Security > Encryption Contexts**.
2. Click **New**.
3. Enter the following:
   - **Name**: enter the text users see when selecting an encryption context.
   - **Encryption key**: do not change this field If you want the instance to randomly generate a key. Otherwise, enter the desired key (exactly 24-characters for Triple DES, or exactly 16-characters for AES 128-bit, or 32-characters for AES 256-bit).
   - **Type**: select **AES 128-bit** for Advanced Encryption Standard, **Triple DES** for Triple Data Encryption Standard, or **AES 256-bit** if your system is configured for it.

   **Warning:** You cannot retrieve this key from the instance, so save it elsewhere before clicking **Submit** if you will need it.

4. Click **Submit**.

   The encryption key itself is encrypted with a key that is stored in the program, not in the database. This prevents other users from copying the key and using it to decrypt data.

5. Navigate to **System Security > Roles** and open the role to associate with the encryption context.

6. Configure the Roles form to add the Encryption context field.

7. Select the encryption context to associate with the role (there can be only one encryption context per role).

8. Click **Update**.

   Users must log out of the instance and log in again to use the encryption context.

**Add an encrypted text field to a form**

Only a user with configuration rights, such as an administrator, can add an encrypted text field to a form. Encrypted text fields are installed with the Encryption Support plugin.

**Note:** When you convert an existing field to an encrypted field, make sure the field length matches the max length of the encrypted field type.

1. Configure a form.
2. Create a new field and set the Type to **Encrypted Text**.
3. Place the field where it should appear on the form.
4. Click **Save**.
Add an encryption context selector

The encryption context selector enables users who have multiple encryption contexts to select a context when entering data.

Only a user with the admin role can activate the encryption context selector.

1. Click System UI > UI Macros.
2. Open the encryption_select UI macro.
3. Select the Active check box.
4. Click Update.
5. Reload the page to see the choice list (if you have more than one encryption context).

The encryption context selector appears in the welcome banner only for users who have multiple roles with different encryption contexts.

Note: The selector appears only for users who have two or more encryption contexts to choose from. Users with either one or no encryption contexts do not see the selector.

Use an encrypted text field

Encrypted text fields operate differently depending on the number of encryption contexts available to the user.

- User with no encryption contexts: The form hides the encrypted field.
- User with one encryption context: The user automatically uses their encryption context with encrypted text fields.

Figure 708: Encrypted text fields 03
• There is no data in the field: The form displays the encrypted field (assuming UI policy does not prevent it). Users with any encryption context can see empty encrypted fields. Entering data in the field causes the encrypted fields to use the currently selected encryption context to encrypt the data.

Figure 709: Encrypted text fields 01
Figure 710: Encrypted text fields 01a

- There is data in the field: If the user has access to the matching encryption context, the form displays the encrypted field.

Figure 711: Encrypted text fields 02
• User with two or more encryption contexts: The user can select an encryption context from the selector in the welcome bar.

• There is no data in the field: The form displays the encrypted field (assuming UI policy does not prevent it). Users with any encryption context can see empty encrypted fields. Entering data in the field causes the encrypted fields to use the currently selected encryption context to encrypt the data.

• There is data in the field: If the user has access to the matching encryption context, the form displays the encrypted field. The encrypted field always uses the original encryption context to encrypt changes to the field. This prevents users with multiple encryption contexts from changing the encryption context of a field.

Note: A lock icon appears next to the field label to indicate an encrypted field. Pointing to the icon displays the name of the context used to encrypt the field.

Figure 712: Encrypted text fields 02a

Figure 713: Encrypted text fields lock icon

Note: Users cannot use encrypted fields in a filter.
Use attachment encryption

You can encrypt attachments that are already attached to records.

1. Log in as a user with at least one encryption context.
2. If more than one encryption context is available, select the encryption context for this session from the selector.
3. Navigate to a form which needs an attachment added, such as the Incident or Problem form, and click the attachment icon to open the Attachments dialog box.
4. Select the file to be attached.
   Only users with one or more encryption contexts see the Encrypt file check box below the file name.
5. Select the **Encrypt file** check box.
   Users with more than one encryption context are asked to confirm the encryption context. If you select a different encryption context, the encryption context selector updates to reflect the change.
6. Click **Attach** to upload the file attachment.
   The file appears in the Current file attachments section of the form with a special icon indicating that it is encrypted. Pointing to the icon shows the name of the encryption context.
7. Click **Done**.
   Attached files are listed across the top of the form. A special icon identifies encrypted files. Note that you can only see the encrypted files for which you have the encryption context.

Encrypt MID Server login credentials

The MID Server login credentials appear in the config.xml file in clear text. If access to the MID Server host machine is not secure, store the login credentials in this file in encrypted form.

Role required: admin

**Note:** Any field in this file can be encrypted, but once encrypted, a field can be managed only from within the instance. The best practice is to encrypt password fields only (for the MID Server and any proxy server specified).

1. Navigate to the agent directory that was created when the MID Server was installed and open the config.xml file using a text editor such as WordPad.
   The instance credentials section of the config.xml file looks like this:

   ```xml
   <!-- If your instance has authentication enabled (the normal case), set these parameters to define the user name and password the MID Server will use to log into the instance. -->
   <parameter name="mid.instance.username" value="midsrvadmin" />
   <parameter name="mid.instance.password" value="securepassw0rd"/>
   ```

2. Add the encrypt="true" attribute to the password tag.

   ```xml
   <parameter name="mid.instance.username" value="midsrvadmin" />
   <parameter name="mid.instance.password" encrypt="true" value="securepassw0rd"/>
   ```

3. Save the config.xml file, and then restart the MID Server service.
The password is now encrypted.

The password cannot be decrypted (or displayed in clear text again) by changing the encryption attribute to false or by deleting the attribute. If the password is changed in the config.xml file and the MID Server restarted, the new password is encrypted.

Encryption Scripting

There are several example scripts for you to use with encryption.

Encrypt Unencrypted Attachments with Script

The following sample script encrypts unencrypted attachments, such as in the incident table.

```javascript
bulkEncryption();

function bulkEncryption() {
  gs.log("*********** BULK ENCRYPTION RUN BY " + gs.getUserName());
  encryptAttachments("incident", "testContext");
  gs.log("*********** BULK ENCRYPTION COMPLETED");
}

// Note that whomever runs this script must have access to use the
// specified encryption context or nothing will happen when
// "changeEncryptionContext" is called except that a warning will appear
// in the log: WARNING *** WARNING *** Attempt to get
// cipher for encryption context 'contextName' without authorization
function encryptAttachments(table, encryptionContextName) {
  var contextGR = new GlideRecord("sys_encryption_context");
  contextGR.addQuery("name", encryptionContextName);
  contextGR.query();
  if (!contextGR.next()) {
    gs.log("*********** No such encryption context " + encryptionContextName);
    return 0;
  }
  var encryptionId = contextGR.getUniqueValue();
  gs.log("*********** BEGIN ENCRYPTING ATTACHMENTS FOR " + table + " TABLE");
  var attachmentGR = new GlideRecord("sys_attachment");
  attachmentGR.addQuery("table_name", table); // only attachments for the
  specified table
  attachmentGR.addNullQuery("encryption_context"); // only attachments not
  yet encrypted
  attachmentGR.query();
  var count = 0;
  while (attachmentGR.next()) {
    var sysAttachment = new GlideSysAttachment();
    sysAttachment.changeEncryptionContext(attachmentGR.getValue("table_name"),
      attachmentGR.getValue("table_sys_id"),
      attachmentGR.sys_id, encryptionId);
    gs.log("*********** ENCRYPTED [" + attachmentGR.sys_id + "] " +
      attachmentGR.getValue("file_name"));
    count++;
  }
  gs.log("*********** ENCRYPTED " + count + " ATTACHMENTS FOR " + table + " TABLE");
}
To write a script changing the encryption context from one context to another, access to both contexts is required.

Set the Encryption Context ID in Script

A function, `setContextID`, is provided to set the encryption context ID of an encrypted field in script. The argument to the function is the `sys_id` of the encryption context you wish to use. The following example creates records with an encrypted field, `u_subspecies`, encrypted by the "Species Security" encryption context.

```javascript
Note: The user who runs this script must have access to the encryption context.
```

```javascript
encrypt();

function encrypt() {
    var encryptionID = getEncryptionID("Species Security");
    if (encryptionID == "")
        return;

    createEncryptedRecord("Sumatran tiger", "Panthera", "tigris", "sumatrae", encryptionID);
    createEncryptedRecord("Bengal tiger", "Panthera", "tigris", "tigris", encryptionID);
    createEncryptedRecord("Siberian tiger", "Panthera", "tigris", "altaica", encryptionID);
}

function getEncryptionID(encryptionName) {
    var contextGR = new GlideRecord("sys_encryption_context");
    contextGR.addQuery("name", encryptionName);
    contextGR.query();
    if (!contextGR.next()) {
        gs.log("*********** No such encryption context " + encryptionContextName);
        return "";
    }
    return contextGR.getUniqueValue();
}

function createEncryptedRecord(commonName, genus, species, subspecies, encryptionID) {
    var newRecord = new GlideRecord("u_species_encrypted");
    newRecord.u_common_name = commonName;
    newRecord.u_genus = genus;
    newRecord.u_species = species;

    // We have one encrypted field, u_subspecies.
    newRecord.u_subspecies.setDisplayValue(subspecies);
    newRecord.u_subspecies.setContextID(encryptionID); // use our encryption context for the field
    newRecord.insert();
}
```

Remove Encryption Contexts

These functions remove an encryption context, or all encryption contexts, from those available to the user session.
This can be used by customers in a script to further limit the encryption contexts that are available to a user under certain conditions. For example, a synchronous Script Action triggered by the `session.established` event could remove all encryption contexts if a user was not connecting from an IP in a certain range.

```javascript
if (some condition) {
    var contexts = gs.getSession().getEncryptionContext();
    contexts.removeAllContexts();
}
```

Individual encryption contexts can be removed as well using the `sys_id` of the encryption context.

```javascript
if (some condition) {
    var contexts = gs.getSession().getEncryptionContext();
    contexts.removeContext("077d9b3307211000e2bb5720e1021e61");
}
```

**Return a Decrypted Field Value**

You can return the field value for an encrypted field by using the `getDisplayValue` method.

**Async business rules**

Async Business Rules actually operate via impersonation and impersonation does not give you the impersonated user's encryption contexts.

Therefore, you will not want to use async Business Rules when you want to encrypt fields.

**Encryption Demo plugin**

A demo plugin (`com.snc.task_encryption.demo`) is available which demonstrates how the encrypted fields and files behavior can be customized using Business Rules, UI Actions, and UI Policy.

The demo forces these forms to use only one security context per record. The first person to add encrypted information to a record "claims" the record for that context. A button is provided to change the context of the record to another one if the user has more than one context.

To try the demo on a demonstration system (not recommended for production instances since it modifies the task table among other things), activate the plugin `com.snc.task_encryption.demo` then add the "social security numbers", "credit cards", and "encryption context fields" to the incident form. If encryption contexts have not been created and allocated to users, follow the set up instructions above.

UI Action button **Add Secure Info** (actually two buttons) appears on records without encrypted data and ask which context to use (if the user has more than one) and unhide the social security numbers and credit card fields.

UI Action button **Change Encryption** appears on records that do have encrypted data for users with more than one context.

The business rule "Form Single Encryption Context" enforces the one context per form rule.

The business rule "Update on New Encrypted Attachment" causes the context of encrypted files to be noted on the corresponding owning record.

UI Policies show the encrypted fields as appropriate.

**Encryption support FAQ**

These are general security FAQs for encryption. For information about MID Server credential encryption.
**What prevents an admin with privileged access from accessing the key?**

A user with access to the encryption context record could access it. The key itself is stored encrypted.

**Where does encryption/decryption occur?**

Encryption and decryption occurs on the server not in the user interface.

**Is the key loaded into memory only when required to encrypt/decrypt?**

Yes, first the user’s access to the key is validated and then created in memory to do the encryption/decryption and then disposed of after the operation is complete.

**Is the key stored in clear-text on the server anywhere? If it is encrypted where is that encryption key stored and how is it protected?**

The encryption context key is stored encrypted in the database. The encryption key for that encryption context key is built into the program and is not visible through the user interface at any time.

**What stops an admin with privileged access from giving themselves the role that is allowed to encrypt/decrypt?**

A user with admin access could grant himself or other users the role associated with the encryption context as this is how encryption contexts are assigned. If desired, additional measures could be added by the customer or professional services such as sending an email to an appointed “encryption manager” whenever a role associated with an encryption context is granted to a user, etc.

**What level of logging is there to detect changes to a role that allows encrypt/decrypt?**

There is no extraordinary logging for changes to roles aside from the logging of the transaction. This certainly could be added via business rules by the customer or ServiceNow professional services in a wide variety of ways depending on the needs of the customer.

**What stops an admin with privileged access from tampering with the logs?**

The product keeps two sets of logs. One is visible within the instance as the “system log”. An administrative user can, in theory, manipulate this log, although the security manager can be configured to make such tampering extremely difficult. A second log exists on the file system of the application server and cannot be manipulated directly from within the App server. In the event of a forensic situation wherein an administrator has deliberately tampered with the application’s own internal auditing and logging capabilities, the file system based log can be used to reconstruct a user’s transactional history.

**Can I encrypt Import Sets?**

*Import Sets* run as user System. Therefore, import sets cannot add data to encrypted fields.
Can I encrypt fields in Inbound Email Actions?

*Inbound email actions* on page 2822 work via impersonation, but impersonation does not have access to users’ encryption contexts. Carefully consider the safety of mailing data worthy of encryption through email routers.

Do templates support encrypted fields?

Templates do not support encrypted fields. Applying a template does not apply data to an encrypted field. Saving a template from a form does not save encrypted values.

Sample Java digest algorithm for encryption

This Java algorithm illustrates creating a digest token from an HTTP header.

This sample assumes:

- The web server supports Java
- The hash computation method is SHA1
- The secret key value is abc123
- The unencrypted HTTP header name is user_name

Change the Java code to use another hash computation mechanism (such as MD5), change the secret key value, or HTTP header name.

```java
import javax.crypto.Mac ; import javax.crypto.spec.SecretKeySpec ; import 
sun.misc.BASE64Encoder ; // public class DigestTest  { private static final 
String MAC_ALG  = "HmacSHA1" ; // default to something JDK 1.4 has String 
fKey  = "abc123" ; public byte [ ] getDigest ( String acct ) { try { byte [ ] bkey  = fKey. getBytes ( ) ; byte [ ] data  = acct. getBytes ( ) ;
Mac mac  = null ; try {
    mac  = Mac. getInstance (MAC_ALG ) ;
    mac. init ( new SecretKeySpec (bkey, MAC_ALG ) ) ;
} catch ( Exception e ) {
    e. printStackTrace ( ) ;
} byte [ ] sig  = mac. doFinal
(data ) ; String signature  = new String (sig ) ; System. out. println
("value:" + acct ) ; System. out. println ( "digested value: "+ signature ) ; return sig ; } catch ( IllegalStateException e ) {
    e. printStackTrace ( ) ;
} catch ( IllegalStateException e ) {
    return null ;
} public static void main
(String [ ] args ) { 
    BASE64Encoder encoder  = new BASE64Encoder ( ) ;
    DigestTest test  = new DigestTest ( ) ; String userName  = "user_name"
; System. out. println ( "base 64 digest username: " + encoder. encode (test.
getDigest (userName ) ) ) ;
}
```

Demonstration plugin

The instance provides a demonstration plugin called Encryption Support - Single Context Task Encryption Demo (com.snc.task_encryption.demo).

The demonstration plugin is meant to illustrate how to use business rules, *UI actions*, and UI policy to support encrypted fields. The plugin enforces a limit of one security context per record. The first person to add encrypted information to a record sets the context.

You should test the demonstration plugin on a sub-production instance since it modifies the Task table and adds business rules and a UI policy. After activating the plugin:
1. Create one or more encryption contexts.
2. Configure the Incident form to add the Social security number, Credit card, and Encryption context fields.

The plugin UI actions are:

- Add Secure Info: appears on records without encrypted data and allows the user to display the social security numbers and credit card fields.
- Add Secure Info: appears on records without encrypted data and prompts the user to select an encryption context (if the user has more than one).
- Change Encryption: appears on records with encrypted data and prompts the user to select an encryption context (if the user has more than one).

The plugin business rules are:

- Form Single Encryption Context: ensures that each form only has one security context.
- Update on New Encrypted Attachment: ensures that each attachment has the same security context as the parent incident.

The plugin UI policies show the encrypted fields when appropriate.

Encrypt a password in system properties

The Encrypt SysProperty Password business rule automatically encrypts the value of any system property with the type password or password2.

The business rule encrypts passwords when you add a new value or update an existing one. After encrypting the password, the instance always stores and displays the encrypted password value. The instance only decrypts the password in memory and never saves a clear-text version.

After upgrading from an earlier version, encrypt existing properties that contain passwords by updating the value.

1. In the Type filter text, enter `sys_properties.list`.
2. Select the system property you want to encrypt.
3. Enter or update the Value to change the password.
4. Click **Update**.

The business rule encrypts the password.
Add an encryption context selector

The encryption context selector enables users who have multiple encryption contexts to select a context when entering data.

Only a user with the admin role can activate the encryption context selector.

1. Click **System UI > UI Macros**.
2. Open the encryption_select UI macro.
3. Select the **Active** check box.
4. Click **Update**.
5. Reload the page to see the choice list (if you have more than one encryption context).

The encryption context selector appears in the welcome banner only for users who have multiple roles with different encryption contexts.

---

**Note:** The selector appears only for users who have two or more encryption contexts to choose from. Users with either one or no encryption contexts do not see the selector.

---

VPN setup to integrate datacenters

Your instance supports direct integrations with any number of data sources over the Internet.

When configuring an integration that uses an encrypted protocol, such as Lightweight Directory Access Protocol (LDAP) or HTTPS, it is good practice to use the Internet as a transport mechanism.

However, there may be security or network architecture requirements that dictate the use of a site-to-site Internet Protocol Security (IPSEC) Virtual Private Network (VPN) connection between the datacenters and your business networks. The VPN supports the necessary encrypted communication between the instance and your network.
This video describes how to locate the IP addresses for each of your company's instances.

Create an address for VPN communication

To prevent conflict or overlap with internal ServiceNow networks or with another customer's internal IP address schemes, the instance requires that all tunneled traffic in the encryption domain use non-RFC-1918 addresses on both sides of the tunnel.

The instance provides a single IP address for the source of queries into your network.

- Provide Network Address Translation (NAT), non-RFC-1918 addresses for each host that is integrating with the instance.

Alternatives to using a VPN

There are alternatives to using a VPN that you might want to consider.

These alternatives provide a simpler way to connect your instance to the resources in the ServiceNow data centers and provide better encryption. Additionally, you can avoid any issues that VPN downtime might cause, such as making your instance unavailable to users if there is an issue with the VPN tunnel.

Single sign-on and MID server

Consider using a combination of Single Sign-On (SSO) for authentication and the MID Server for user data synchronization, rather than using a VPN to connect your LDAP server to your instance. For integrations other than LDAP, consider using certificate-based encryption.

Starting with the Eureka release, you can use:

- The Okta for single sign-on (see OKTA SSO integration on page 1945) for more information.
- The LDAP listener on a MID server to synchronize your user table in near real time.

The advantage of this approach is that there are no firewall holes, routes, VPN tunnels, or other special network settings to configure and maintain. The SSO/MID-Server solution is the most flexible, secure and cost-effective method to achieve the complete LDAP integration.

LDAP over SSL

Another alternative to using a VPN tunnel is to configure LDAP Over SSL (LDAPS) directly over the Internet. You can configure a read-only domain controller and lock the instance down in your DMZ using only the instance's source addresses and the destination ports of your choice. Since the ports for LDAP are configurable in your instance, you can perform a port address translation (PAT) if desired. With LDAPS, you control the certificate that is uploaded over an encrypted channel to the instance, (see Upload a certificate to an instance on page 2450). The packets cannot be encrypted or decrypted without the certificate.

The advantage of this approach is that it provides a stronger encryption and decryption mechanism. A VPN can only encrypt and decrypt the traffic between the two peers sitting on the Internet with a coordinated pre-shared key, similar to a password. LDAPS provides a longer encrypted path, end-to-end, at the application layer and with a certificate that is far more complicated than a pre-shared key that the IPSec tunnel uses.
Addresses for VPN communication

To prevent conflict or overlap with internal ServiceNow networks or with another internal IP address schemes in your network, all tunneled traffic in the encryption domain must use non-RFC-1918 addresses on both sides of the tunnel.

ServiceNow provides a single IP address for the source of queries into your network. You must provide Network Address Translation (NAT), non-RFC-1918 addresses for each host that is integrating with your instance. These public addresses need to be owned by your organization. Third party addresses cannot be used inside tunnels. Additionally, the encryption domain must not contain the IP address of the VPN peer.

Redundant tunnels

If you are using a VPN, you can create redundant tunnels.

There are two ways to build redundancy for your tunnels:

- Using the same encryption domain behind both of your peers. This is the preferred method.
- Using a different encryption domain behind each peer.

With the first method, you need to provide the same NAT address behind each of your peers to create a connection path using that address to your server. The path to your server could be the same physical machine or a mirror which provides identical services. With this method, your instance would use the same IP address to connect to your servers regardless of whether your primary or secondary tunnel is active. If you have more than one server, follow this same scheme for your additional servers. This method provides the most transparency to your users and is recommended.

The second method requires configuration in your instance to provide the redundancy. When the tunnel is used for LDAP, for example, you could provide redundant LDAP servers in your instance. Note that this method requires the connection to the first configured LDAP server to timeout before the instance attempts to connect to the secondary server. Because of this additional time delay, this solution should only be implemented if the first option is unattainable. Also note that not all services can be configured for redundantly in your instance. If you are using a VPN tunnel for something other than LDAP and redundancy is required, check that your configuration can support multiple addresses, or see the first option above.

ServiceNow VPN connections

The ServiceNow VPN infrastructure uses pairs of Cisco adaptive security appliance (ASA) devices that serve as VPN termination points.

The VPN between the instance and your network utilizes your existing networking hardware to support communications. It is not necessary to install a piece of hardware. Because each customer has a unique configuration, the instance has a flexible VPN solution. the instance has built tunnels to Checkpoint, Juniper, Nortel, and other IPSEC VPN-capable devices.

The VPN connections between the instance and your network are created to support the encrypted flow of traffic into your network. Frequently, integrations that use the VPN do not have encryption as part of the underlying protocol. For example, LDAP over the VPN versus LDAPS over the Internet and HTTP over the VPN versus HTTPS over the Internet.

The network does not allow any inbound-to-ServiceNow integration or end-user-to-ServiceNow traffic to traverse a VPN connection. This restricted communication includes end-user access to the platform, administration of the platform, web services integrations, and other integrations that are configured to use a MID Server. All such inbound communication to the instance must be performed over the Internet using HTTPS. This configuration provides an encrypted communication channel. The encryption channel, along with IP access control, meets the security requirements for this traffic flow.
VPN setup

From the time that a VPN request is submitted, it typically takes one week or less to complete the VPN build.

To support the redundancy requirements of your instance and your organization, a minimum of two and a maximum of four VPNs are provisioned (from the active site to your active site or the active site to your DR site, and so on).

It is good practice for the encryption domain to be as specific as possible. Ideally, the encryption domain would include only the specific hosts that are required for the integrations. A large encryption domain can create opportunities for routing discrepancies (VPN versus Internet).

To create the VPN, the instance does the following:

- Provides the VPN peer and host addresses from each datacenter.
- Builds the necessary VPN connectivity from two datacenters into your network. To support redundancy and disaster recovery (DR) requirements, the VPNs can be provisioned from two datacenters into two networks.

The instance does not support building multiple VPN tunnels into a customer network for the purpose of connecting to multiple geographic regions or subsidiaries. You should perform any inter-site routing, traffic distribution, or traffic shaping within your own internal network, rather than having multiple VPN tunnels.

Request a VPN service

For all VPN requests, including provisioning, modifications, or general questions, use the Service Catalog VPN Request form.

1. Open HI.
2. Navigate to Self-Service > Service Requests > VPN Requests.
3. Select the appropriate VPN request type.
4. Answer the questions.
   - Questions vary depending on the request type selected.
5. Click Submit.

Once your request is submitted, ServiceNow will work with your network engineer(s) to test and validate that the VPN is successfully passing traffic. To ensure that your questions are answered in a timely manner, please address VPN-related questions during this process.

HTML sanitizer

The HTML sanitizer automatically cleans up HTML markup in HTML fields and translated HTML fields to remove unwanted code and protect against security concerns such as cross-site scripting attacks.

The HTML sanitizer works by checking the built-in white list for markup that you always want to preserve. The sanitizer provides the HTMLSanitizerConfig script include that administrators can use to modify the built-in white list. Items can also be added to the black list, which overrides the white list, to remove HTML markup.

The following types of items can be added to white and black lists:

- Global attributes
- Any HTML elements

**Note:** By default, URL attributes like `href` and `src` support only these protocols:
• http
• https
• mailto

For example:

<a href="https://community.servicenow.com/welcome">Community</a>

The Default White List

BUILTIN_HTML_WHITELIST :

    globalAttributes: { attribute: ["id","class","lang","title","style"],
                        attributeValuePattern: {}},

    label: { attribute: ["for"],

    font: { attribute: ["color","face","size"],

    a: { attribute: ["href","nohref","name","shape"],

    img: { attribute: ["src","name","alt","border","hspace","vspace","align","height","width"],

    table: { attribute: ["border","cellpadding","cellspacing","bgcolor","background","align","noresize","height","width","summary","frame","rules"],

    th: { attribute: ["background","bicolor","abbr","axis","headers","scope","nowrap","height","width","align","char off","char","colspan","rowspan"],

    td: { attribute: ["background","bicolor","abbr","axis","headers","scope","nowrap","height","width","align","char off","char","colspan","rowspan"],

    tr: { attribute: ["background","height","width","align","valign","char off","char"]},

    thead: { attribute: ["align","valign","char off","char"]},

    tbody: { attribute: ["align","valign","char off","char"]},

    tfoot: { attribute: ["align","valign","char off","char"]},

    colgroup: { attribute: ["align","valign","char off","char","span","width"]},

    col: { attribute: ["align","valign","char off","char","span","width"]},

    p: { attribute: ["align"]},

    style: { attributeValuePattern: {"type":"text/css"} }

    canvas: { attribute: ["height","width"]},

    details: { attribute: ["open"]},

© 2017 ServiceNow. All rights reserved.  2549
Activate HTML sanitizer

The HTML sanitizer provides a property to enable or disable the sanitizer for all HTML fields in the system.

Role required: admin

By default, the property is set to true for new instances.

1. In the navigation filter, enter `sys_properties.list`. 
2. Set the properties glide.html.sanitize_all_fields and glide.translated_html.sanitize_all_fields to true. If the properties do not exist in the System Properties table, you can add them.

**Configure HTML sanitizer**

You must modify a script include to make configuration changes to the HTML sanitizer.

Role required: admin

1. Navigate to **System Definition > Script Includes**.
2. Open **HTMLSanitizerConfig**.
3. To add items to the black list, use the HTML_BLACKLIST class. To add items to the white list, use the HTML_WHITELIST class.

Use this format:

```
HTML_XXXXLIST :{
  globalAttributes :{
    attribute: [attribute-name1,...],
    attributeValuePattern: {attribute-name2:attribute-value-regex-pattern,...}
  },
  <html-element-name> : {// Same as Above}, ----
}
```

- **globalAttributes** contains attribute or attributeValuePattern items that are applicable globally for all the HTML elements.
- **attribute** is a comma-separated list of attributes.
- **attributeValuePattern** is a dictionary of attribute to attribute-value-regex-pattern pairs. The attribute-value-regex-pattern is a regular expression which has to match the attribute value.

Consider the following example:

```
HTML_WHITELIST :{
  globalAttributes :{
    attribute: ["id","name"],
    img :{
      attribute: ["style","align"],
      attributeValuePattern : {src:".*jpeg"},
      iframe : {}},
}
```

It adds the following items to the white list:

- The global attributes id and name. This is a list of strings that can be applied globally to all the elements.
- The img element where the attributes are style and align.
- The img element where the source attribute of the image is a file with the .jpeg extension. This is an example of a regular expression pattern that matches an attribute value.
- The iframe element.

**Enable sanitization on individual fields**

You can use field attributes to enable or disable the sanitizer on individual fields.
Role required: admin

You need to first set the sanitizer property to false, and then enable the sanitizer on a per-field basis for any form.

1. Navigate to the sys_properties table and set the glide.html.sanitize_all_fields to false. This disables the sanitizer for all HTML fields in the system.
2. Navigate to the form that contains the HTML field.
3. Right-click the HTML field label, and select Configure Dictionary. The Dictionary Entry form opens for the HTML field.
4. Enter one of the following in the Attributes field:
   - To disable sanitization enter html_sanitize=false
   - To enable sanitization enter html_sanitize=true
5. Click Update.
6. To enable the HTML sanitizer for translated HTML fields, set the glide.translated_html.sanitize_all_fields property is true.

Enable HTML Sanitizer logging

When the HTML sanitizer removes elements or attributes, they are added to the system log.

Role required: admin

You can review these sanitized elements by adding /syslog_list.do?sysparm_query=source %3DHTMLSanitizer to your instance URL.

1. To review these sanitized elements add /syslog_list.do?sysparm_query=source %3DHTMLSanitizer to your instance URL.
2. To enable or disable logging, add the glide.html_sanitize.discarded_log.enable property to the system properties and set the value to true (enabled) or false (disabled). This property is true by default.

System logs

System logs display warnings and errors within the instance processes and records, and non-critical events such as memory usage on the server machine.

This list view displays the log entries for the current day only. To view other log files, use the log file browser.

This log provides the following information for all occurrences:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the logging activity for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Level</td>
<td>Type of message displayed. The levels are Debug, Error, Warning, and Information. A warning is an error that has been handled and recovered. An error is something that must be fixed.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Message</td>
<td>System-generated message regarding the nature of the occurrence.</td>
</tr>
<tr>
<td>Source</td>
<td>Name of the process or area affected by the occurrence. For example, the source of the occurrence might be EMAIL or Memory.</td>
</tr>
</tbody>
</table>

**Transaction logs**

The transaction log records all browser activity for an instance. This log provides the following information for all activities.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the browser action for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Created by</td>
<td>The user who created this activity.</td>
</tr>
<tr>
<td>Response time</td>
<td>Round trip response time for the browser request, in milliseconds.</td>
</tr>
<tr>
<td>Network time</td>
<td>Latency time of the network response after the browser request is made, in milliseconds.</td>
</tr>
<tr>
<td>Output length</td>
<td>Size of the output string sent by the instance to the browser, in bytes.</td>
</tr>
<tr>
<td>SQL count</td>
<td>Number of SQL server commands executed for this activity.</td>
</tr>
<tr>
<td>Business rule count</td>
<td>Number of business rules executed for this activity.</td>
</tr>
<tr>
<td>Business rule time</td>
<td>Elapsed time for the execution of the business rules for this activity.</td>
</tr>
<tr>
<td>URL</td>
<td>The application or module connected to by the client browser.</td>
</tr>
<tr>
<td>System ID</td>
<td>System generated identifier of the client instance making the request. This ID is used for cluster environments in which several instances (nodes) communicate with the database.</td>
</tr>
<tr>
<td>IP address</td>
<td>IP address of the client making the request.</td>
</tr>
<tr>
<td>GZipped</td>
<td>Indication of whether a compressed Web page was requested by the browser.</td>
</tr>
<tr>
<td>Protocol</td>
<td>The HTTP protocol used by the browser for this instance.</td>
</tr>
</tbody>
</table>

The **System Scheduler > Slow Job Log** module provides a transaction log filtered to show only slow transactions.
Client transaction timings

The Client Transaction Timings plugin enhances the system logs by providing more information on the durations of transactions between the client and the server.

By providing information on how time was spent during the transaction, performance issues can be tracked down to the source by seeing where the time is being consumed.

This plugin requires the Response Time Indicator to be enabled, and collects information from the following browsers:

- Firefox
- Internet Explorer
- Chrome

Client Transactions Information

Installing the plugin adds the module Client Transactions to the System Logs application, which provides a list of every logged transaction between client and server within the last day. The following information is tracked:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>The moment the transaction was recorded.</td>
</tr>
<tr>
<td>Response Time</td>
<td>The number of ms spent by the server in fulfilling the transaction.</td>
</tr>
<tr>
<td>Business Rule Time</td>
<td>The number of ms spent by business rules triggered by the transaction.</td>
</tr>
<tr>
<td>SQL Time</td>
<td>The number of ms spent by the SQL database.</td>
</tr>
<tr>
<td>Client Response Time</td>
<td>(Load_completion_time) - (start_time). It is inclusive of server time.</td>
</tr>
<tr>
<td>Client Network Time</td>
<td>The number of ms spent by the network the client is connecting through.</td>
</tr>
<tr>
<td>Browser Time</td>
<td>The number of ms spent by the browser during the transaction.</td>
</tr>
<tr>
<td>Client Script Time</td>
<td>The number of ms spent executing client scripts</td>
</tr>
<tr>
<td>UI Policy Time</td>
<td>The number of ms spent executing ui policy</td>
</tr>
<tr>
<td>Type</td>
<td>Type of transaction (one of Form, List, Other)</td>
</tr>
<tr>
<td>Table</td>
<td>The table that was displayed e.g. incident, change_request</td>
</tr>
<tr>
<td>View</td>
<td>The view for this form/list</td>
</tr>
</tbody>
</table>

Client Detailed Information

A more detailed breakdown of the client timings for all Form rendering (but not list rendering) is also tracked. To see details, drill into a particular client transaction record and observe the related list at the base of the screen.
Table 601: Client Detailed Information

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>The order during the load that this operation occurred</td>
</tr>
<tr>
<td>Type</td>
<td>The type of operation</td>
</tr>
<tr>
<td>Name</td>
<td>Descriptive name of this particular operation</td>
</tr>
<tr>
<td>Duration</td>
<td>Number of ms this particular operation took to complete</td>
</tr>
</tbody>
</table>

Email logs

The email log records all email notifications sent from all instances within the system.
This is a verbose and unfiltered view of email. For a more detailed view, see the System Mailbox application.
This log provides the following information for all notifications.

Table 602: Email log

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox</td>
<td>The system mailbox to use for filtering the email notifications displayed.</td>
</tr>
<tr>
<td>State</td>
<td>The current state of the notification (Error, Ignored, Processed, or Ready).</td>
</tr>
<tr>
<td>Receive type</td>
<td>The type of inbound email notification (None, Forward, New, or Reply).</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>The status of the email notification. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• received: The server received this email.</td>
</tr>
<tr>
<td></td>
<td>• received - ignored: The server received this email, but it was ignored by</td>
</tr>
<tr>
<td></td>
<td>the instance for inbound email action purposes. Typically, these emails are</td>
</tr>
<tr>
<td></td>
<td>either spam or auto-replies. See the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - failed: The server has attempted to send the email and failed. See</td>
</tr>
<tr>
<td></td>
<td>the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - ignored: The server skipped sending this email. Typically, this is</td>
</tr>
<tr>
<td></td>
<td>for an email which was generated but lacked a recipient email address or is</td>
</tr>
<tr>
<td></td>
<td>a duplicate email. See the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - ready: The email is ready to be sent, but has not been sent out by</td>
</tr>
<tr>
<td></td>
<td>the mail server. Typically, an email remains in this state for only a short</td>
</tr>
<tr>
<td></td>
<td>time.</td>
</tr>
<tr>
<td></td>
<td>• sent: The email was sent by the instance without any errors or issues.</td>
</tr>
<tr>
<td>Target</td>
<td>A Document ID reference to the record if the email is generated by an insert,</td>
</tr>
<tr>
<td></td>
<td>update, or delete of a particular record.</td>
</tr>
<tr>
<td>User</td>
<td>The name of the user, from the user record, of the instance from which the</td>
</tr>
<tr>
<td></td>
<td>email notification was sent.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This is a string field.</td>
</tr>
<tr>
<td>Notification Type</td>
<td>The type of notification. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• None</td>
</tr>
<tr>
<td></td>
<td>• SMS</td>
</tr>
<tr>
<td></td>
<td>• SMTP</td>
</tr>
<tr>
<td>UID</td>
<td>The unique ID for the server.</td>
</tr>
<tr>
<td>Created</td>
<td>The date and time of the email activity for the locale of the machine running</td>
</tr>
<tr>
<td></td>
<td>the instance.</td>
</tr>
<tr>
<td>Deleted</td>
<td>An indication of whether the email was deleted from an instance mailbox.</td>
</tr>
<tr>
<td>Weight</td>
<td>The weight of the email, which determines the sending priority relative to</td>
</tr>
<tr>
<td></td>
<td>other notifications on the same table.</td>
</tr>
<tr>
<td>Importance</td>
<td>An indication that the email was sent with a changed level of importance,</td>
</tr>
<tr>
<td></td>
<td>such as Urgent.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Originating Event and Notification</td>
<td>An embedded list that stores the event and notification that initiated the email notification. For more information, see <em>Events</em> on page 2996.</td>
</tr>
<tr>
<td>Subject</td>
<td>A configured description of the action that generated the email notification. You create the subject text for notifications in <em>System Notification &gt; Email &gt; Notifications</em>.</td>
</tr>
<tr>
<td>Error String</td>
<td>The error string captured from the email server to determine why the email was not sent. This is logged only if the email is send-failed.</td>
</tr>
<tr>
<td>Recipients</td>
<td>The email address of the recipient of each notification.</td>
</tr>
<tr>
<td>Body</td>
<td>The body of the email, displayed in raw HTML markup. Use the related link Preview HTML Body to see the body text as rendered HTML.</td>
</tr>
<tr>
<td>Content type</td>
<td>The email content type.</td>
</tr>
<tr>
<td>Headers</td>
<td>Any headers embedded in the email.</td>
</tr>
</tbody>
</table>

Invalid email addresses that the instance strips out of outbound email messages are logged, starting with the Geneva release.

**Outbound email notification recipients**

For outbound notifications, the email system log provides reasons that recipients were included or excluded.

Each log entry corresponds to a reason for inclusion or exclusion. For example, all users who were excluded because they are inactive appear in a single log entry.

A series of system properties can be used to fine-tune the information to be logged. Two master switch properties, `glide.notification.recipient.include_logging` and `glide.notification.recipient.exclude_logging`, control all recipient inclusion and exclusion logging. Several other properties allow you to tailor the information reported in the logs to meet your needs. All of the properties are enabled by default.
Event logs

The event log records all system events that occur within the system.

This log provides the following information for all events that occur:

Table 603: Event log

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the event for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the event. You configure events in <strong>System Definition</strong> &gt; <strong>Business Rules</strong>.</td>
</tr>
<tr>
<td>Parm1</td>
<td>Event-specific value that depends on the event and the recipient.</td>
</tr>
<tr>
<td>Parm2</td>
<td>Event-specific value that depends on the event and the recipient.</td>
</tr>
<tr>
<td>Table</td>
<td>Database table acted on for this event.</td>
</tr>
<tr>
<td>Processed</td>
<td>Date and time the event was processed This time reflects the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Processing time</td>
<td>Time taken to process this event, in milliseconds.</td>
</tr>
<tr>
<td>Queue</td>
<td>Processor queue name.</td>
</tr>
</tbody>
</table>

Import logs

The import log displays information in a verbose format about any data import activity within the platform.

For a more detailed view of the import sets that produced a particular log, see **Import Sets** > **Transform History**.

This log provides the following information for all imports:
Table 604: Import log

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the import for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Level</td>
<td>Type of message displayed. For import files, the level is Information.</td>
</tr>
<tr>
<td>Message</td>
<td>System-generated message regarding the status of the import.</td>
</tr>
<tr>
<td>Source</td>
<td>Name of the external source of the import, such as an integration.</td>
</tr>
</tbody>
</table>

**Workflow logging**

Certain workflow information is tracked in the system logs.

- Each activity executed, including:
  - Date and time started
  - Date and time ended
  - State, for example, Finished, Cancelled, Timed Out, Error
  - Result
  - Fault description, if there was an error

- Transition history, including:
  - Time of transition
  - Activity transitioned from
  - Activity transitioned to
  - Which transition was triggered

- Workflow log, including any log statements added to the workflow

**Configuration logging**

Certain configuration changes are tracked in the system logs.

- Action taken, including insert, update, and delete
- Category of change
- Comments recorded with the change
- Name of the change
- XML difference of the change
- Update set the change is associated to
- Date and time of the change
- User who made the change
- Table where the change was made
- Name of the object being changed
- Type of object being changed
- View the change was made in, for changes to forms or lists
Log utilities

The instance provides the utilities log file browser and log file download.

Use **System Logs > Log File Browser** to view any system log entry. You can search for log files by using the following filters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start time</td>
<td>Start date and time of the range you want to search, for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Session ID</td>
<td>System-generated hexadecimal string that identifies the session that generated the log entry.</td>
</tr>
<tr>
<td>End time</td>
<td>End date and time of the range you want to search, for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Message</td>
<td>System-generated description of the occurrence.</td>
</tr>
<tr>
<td>Level</td>
<td>Type of message displayed. The levels are Debug, Error, Warning, and Information. A warning is an error that has been handled and recovered. An error is something that must be fixed.</td>
</tr>
<tr>
<td>Thread name</td>
<td>System-generated identifier of the thread that created the log file.</td>
</tr>
<tr>
<td>Max rows</td>
<td>Maximum number of records returned for a particular filter.</td>
</tr>
</tbody>
</table>

The instance creates compressed archives of system logs every two days and purges log archives after 21 days. You can download log file archives and view them with **System Logs > Log File Download**. Select a log archive from the list, and then click **Download log** under Related Links to open or save the archive.

**Note:** Log files are only available for the node you are currently logged into. To see the currently logged into node, navigate to **System Diagnostics > Stats**.

Log history

The system uses table rotation and table extension to archive older logs.

By default, the system uses the following schedule to archive common logs:

<table>
<thead>
<tr>
<th>Table</th>
<th>Archive schedule</th>
<th>Rotations</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event [ecc_event]</td>
<td>Every day</td>
<td>7</td>
<td>Rotation</td>
</tr>
<tr>
<td>Queue [ecc_queue]</td>
<td>Every day</td>
<td>7</td>
<td>Rotation</td>
</tr>
</tbody>
</table>
## System diagnostics application

The System Diagnostics application provides logs that relate to the platform.

These logs are available:

- **Upgrade History**: tracks every upgrade to an instance.
- **Slow Queries**: provides insight into how queries affect platform performance. See *Slow Query Logs*.

## Customer updates table

Every change that is made in the system is recorded on the Customer Updates [sys_update_xml] table chronologically.

To navigate to this table, enter `sys_update_xml.list` into the navigation filter.

The following information is stored about each update:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name that identifies the updated record.</td>
</tr>
<tr>
<td>Created</td>
<td>The date and time the customer update record was created.</td>
</tr>
<tr>
<td>Created By</td>
<td>The user who performed the change.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the update.</td>
</tr>
<tr>
<td>Updated</td>
<td>The date and time the customer update record was updated.</td>
</tr>
<tr>
<td>Updated By</td>
<td>The user who performed the update.</td>
</tr>
<tr>
<td>Updates</td>
<td>The number of times the record has been updated.</td>
</tr>
<tr>
<td>Target Name</td>
<td>The name of the element that was altered.</td>
</tr>
<tr>
<td>View</td>
<td>The view of the form that was altered if it was a form layout change.</td>
</tr>
<tr>
<td>Payload</td>
<td>The XML contents of the record after the change.</td>
</tr>
<tr>
<td>Remote Update Set</td>
<td>A reference to that update set if the change was performed by a remote update set.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Local Update Set</td>
<td>The update set the change is associated with.</td>
</tr>
</tbody>
</table>

### Email logs

The Emails module links to all of the mail received or sent by the platform.

For more information, see *System Mailboxes*.

### Audited tables

If a field on a particular table is audited, all changes to that field are tracked.

For more information, see *Turning on Auditing (History) for a Table*.

This information is kept in two places:

- The *History sets* on page 2564 table.
- The *Audit table*.

### Understanding the Sys Audit table

The system tracks inserts and changes to audited records in the *Sys Audit* table.

The system audits tables where the *Audit* check box is selected on the Dictionary record. By default, the system does not audit records from system tables, such as update set tables.

**Note:** To prevent performance issues and infinite loops, the system skips any business rule or workflow triggered by inserts to the Sys Audit table.

### Sys audit record information

Information about a change is stored in a Sys Audit record.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tablename</td>
<td>Table that the audit record is for (for example, &quot;incident&quot;)</td>
</tr>
<tr>
<td>fieldname</td>
<td>Column in the table that the audit record is for (for example, &quot;assigned_to&quot;)</td>
</tr>
<tr>
<td>documentkey</td>
<td>sys_id (Unique Record Identifier) of the record the audit record is for.</td>
</tr>
<tr>
<td>oldvalue</td>
<td>Old value of the field change represented by this sys_audit record.</td>
</tr>
<tr>
<td></td>
<td>- Reference fields: Displays the unique sys_id value of the changed record.</td>
</tr>
<tr>
<td></td>
<td>- Date and time fields: Displays the value in Coordinated Universal Time (UTC) as stored in the database.</td>
</tr>
<tr>
<td>newvalue</td>
<td>New value of the field change represented by this sys_audit record.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
| | • Reference fields: Displays the unique sys_id value of the changed record.
| | • Date and time fields: Displays the value in Coordinated Universal Time (UTC) as stored in the database.

View the structure of the sys audit table
You can view the sys audit table from the system definition.

1. From the left navigation pane, select **System Definition > Tables and Columns**.
2. Scroll to **Sys Audit**.
3. Click the table name.
   The Column names and Column attributes are displayed in the second and third pane.

Enable auditing for a table
You can enable table auditing to track all changes to the table's fields.

Role required: admin

1. Navigate to **System Definition > Dictionary**.
2. Select the table to audit. For example, [cmdb_ci_computer].
3. Select the dictionary entry for the table. The table name always has an empty column name and the type of collection.
4. Check the **Audit** check box.
5. Click **Update**.

   **Note:** By default, the system does not audit records from system tables. To audit a system table, add it to the list of tables in the glide.ui.audit_deleted_tables property list.

Tables do not inherit the audit flags from parent or child audited tables. For example, if you enable auditing for the [cmdb_ci] table, only CIs which are stored in that base table will be audited. Likewise, if you enable auditing for the [cmdb_ci_computer] table, only the computer CI records will be audited, including any fields on the [cmdb_ci_computer] table that are inherited from [cmdb_ci] table.

Deletion of an individual record from a form is audited by default. To prevent auditing, set the table's dictionary attribute **no_audit_delete**.

Deletes from a list are audited when **audit** is checked on the table dictionary and the table is not listed in the property glide.db.audit.ignore.delete.

**Audited table information**
Auditing tracks the following record changes.

- The Unique Record Identifier (sys_id) of the record that changed
- The field that changed
- The new field value
- The old field value
- The number of times this record and field have been updated
- The date and time when the change occurred
- The user who made the change
- The reason for the change (if any reason is associated with the change)
- The record's internal checkpoint ID (if the record has multiple versions)
**Exempted table information**

Some updates are not audited despite enabling auditing on a table. This is why you may see 132 updates in a record's history, but only seven audited ones.

Auditing excludes the following information:

- Any updates made by an upgrade.
- Any updates made through import sets.
- Any records in parent or child tables.
- Any field with the `no_audit` dictionary attribute.
- Any system tables not listed in the `glide.ui.audit_deleted_tables` property list.
- Any field that begins with the `sys_` prefix (system fields) except the `sys_class_name` and `sys_domain_id` columns.
- Any time an inactivity monitor touches a record (this prevents you seeing possibly hundreds of updates listed against an incident, with the noise drowning out the useful data).

**Exclude a field from being audited**

You can prevent a field from being audited by adding the `no_audit` dictionary attribute.

Role required: admin

1. Navigate to **System Definition > Dictionary**.
2. Select the table containing the field you want to exclude from auditing. For example, `cmdb_ci_computer`.
3. Select the dictionary entry for the field you want to exclude from auditing. For example, `hardware_status`.
4. In the Attributes field, enter `noAudit`.

**Table auditing history**

The system tracks incident, change, and problem history in the `[sys_audit]` table.

Enabling auditing tracks the creation, update, and deletion of audited records. In addition, auditing activates the History context menu entry. It is not possible to audit individual fields without auditing the entire table. It is possible, however, to hide certain fields from the audit using a dictionary attribute. Auditing certain system tables that receive a large amount of traffic, such as Workflow Contexts `[wf_context]`, can impact performance and is not recommended.

**History sets**

History sets provide a table for tracking an audited record's history.

Each record's History Set will be generated when the record is inserted, if the record is on an audited table. A user must view a record for the system to create or update a history set for that record.

Several fields of information are captured in the History Set record, displayed in the list view.

### Table 608: List View Record Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>An Document ID for the record whose history is being recorded.</td>
</tr>
<tr>
<td>Table</td>
<td>The audited table for the record whose history is being recorded.</td>
</tr>
</tbody>
</table>
Table 609: Audit History Record Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Time</td>
<td>The amount of time it took to generate the history set.</td>
</tr>
</tbody>
</table>

Field | Input Value |
------|-------------|
Label | The label of the field which was changed.                                  |
Old   | The value before the change.                                               |
New   | The value after the change.                                                |
Type  | Indicates if the entry is for a normal field, an email record, or a relationship change record. |
Update Number | The number of times this field has been changed.                           |
Update Time | The date and time of the change                                           |
User Name | The name of the user who created the change.                               |

History Sets in a Calendar View

Once History Sets are active, the context menu choice History will populate using information from the History Set, rather than from the sys_audit table.

From the user's perspective, the same historical data is available in the same user interface, but the way the information is stored is different.

The History view includes a calendar view, but does not use the normal list interface to filter and interact with the history records. This allows:

• Searching and filtering historic data.
• Exporting historic data.

Viewing history sets

There are two ways of viewing the history set, accessible through the Context Menu action History.

Differences Between Audit and History Sets

The Audit [sys_audit], History Sets [sys_history_set], and History [sys_history_line] tables store the same data, but they serve different purposes and manage data differently.

The Audit [sys_audit] table is where the system stores historical information for all records. These records are intended to be kept forever so that administrators can always track the history of audited records. As the number of auditing records grows over time it becomes more and more inefficient to directly query the Audit table for historical information. It is much more efficient to run queries only on the smaller subset records you actually want to view historical information for.

The History Set [sys_history_set] table identifies which particular records from an audited table have historical information. The History [sys_history_line] table stores the actual changes to field values that
occurred. The system automatically generates History Set and History records as needed from the Audit table when a user either creates a record or requests its history. Rather than containing a complete history of all changes in the system, History Set and History records only contain a recent subset of historical information for records where users have created or requested such information.

The system limits History Set and History records by:

- Having the table cleaner delete History Set records that have not been updated in 30 days.
- Using table rotation to rotate between four History tables every seven days. This means the system drops History records that are older than 28 days.

Should someone need historical information again at a later date, the system can regenerate it from auditing source records.

After the system generates History Set records, the context menu choice History uses the History Set rather than Audit records. From the user's perspective, the same historical data is available in the same user interface, but the way the information is stored is different.

**Tracking changes to reference fields**

Administrators can track changes to reference field display values.

Since reference fields only store an ID value, the system can normally only audit changes when the ID value changes. By default, the system does not audit changes when a reference field display value changes.

Consider the following situation. A user changes her name from Jane Smith to Jane Miller. Since the user name is the display value for the User table, any previous reference to Jane Smith instead refers to Jane Miller. If the administrator just updates the name of the existing user record, audit and history records will only display the new name Jane Miller. By default, the system does not provide a way to distinguish between changes made under the original user name versus those made with the new user name.

If your auditing policy requires tracking user name changes, you can:

- Create a new user record for the new name and deactivate the previous user record. The system preserves audit records for the old user name and creates future audit records with the new user name.
- Create custom fields and a business rule to save the previous name and the date of the name change. The system can use this information to construct the proper names in audit and history records.

**Tracking inserts**

By default, the system does not create Audit records for inserts because in a typical instance, inserts can account for over 80% of the size of the Audit table.

Not tracking inserts allows for better performance and a much smaller Audit table. Administrators can enable auditing of inserts by setting the glide.sys.audit_inserts property to true.

**View history of a CI**

When auditing is enabled for a table that you have access to, you can see a full history of changes for any record in the table. You can view the history of changes in a calendar or list format. For a CI, you can also view its history in a timeline format.

Role required: admin

1. Right-click the form header, and then click History.
2. Select a format for the history.
   
   You can select Calendar or List. On a CI form, you also have the Timeline option.
CI Relationships and History

Changes to a CI relationship (CI Relations, CI/User Relations, or CI/Group Relations) appear in the history of the items on both sides of the changed relationship regardless of whether the change was manual or a result of Discovery.

For example, if the computer alpha has a used by CI Relation with the computer beta, then the history for alpha will have a record of when the relationship with beta was established, and likewise, the history for beta will have a record of when the relationship with alpha was established. The screenshot below shows an example of the history displayed when some CI Relations are established, and when one of them is removed:

![Figure 716: CI Relationship History](image)

The created bullet indicates the date that the CI, user, or group was created. The last activity bullet refers to when the relationships were last changed. If you don't want to show CI relationship history for any or all CI relationship types, you can turn it off by disabling auditing on the CI relationship tables (CI Relationship [cmdb_rel_ci], CI/User Relationship Type [cmdb_rel_user_type], or Group Relationship [cmdb_rel_group]).

Control access to history

You can give a role access to view audit history by setting a system property.
Role required: admin

1. Navigate to **System Properties > System**.
2. In the property List of roles (comma-separated) that can access the history of a record, enter the user roles you want to access history.
3. Click **Save**.

Any changes to a field are omitted if a user without read-access views the history of a record.

**History List**

The history list displays each change as its own row in the change list.
Figure 717: View History List

Click on a row item to view additional details about the change.
To view a history list, the following requirements must be met.

- **Auditing**: Auditing for the table must be enabled to view a history list.
- **ACLs**: By default, the List history option is only available to users with the admin user role. To enable this option to non-admins, create a custom ACL rule granting read access to the Record History [sys_history_set] table.
- **Roles**: At least one of the roles that the user has must be included in the glide.history.role property, which includes the itil role by default.

**Change the number of history entries**

By default, the history displays a maximum of 250 history entries, but you can change this value.
Role required: admin

1. Navigate to System Properties > System.
2. In the property Maximum number of field entries displayed in record history, default is 250, enter a new maximum number of entries.

History Calendar

The history calendar shows you the days where the record was changed, who made the change, and when.

Each user is assigned a color so you can tell at a glance the times a record was changed by a particular person. For example:

Figure 719: Changes Made by Sys Admin

Figure 720: Changes Made by ITIL User

To highlight changes to a particular field, select the field from the Highlight changes to field selection box. Picking a field from this selection box changes the calendar to highlight the times that field was changed. Hover over the text of one of a highlighted change to see the change in value.
Incident History Detail

Details for INC0000039

Created 2012-04-05 17:42:29 by admin
Last updated 2012-06-15 12:56:25 by itil
Update count 3 (1 audited)

2012-04-05 17:42:29 Created by System Administrator (70 Days 19 Hours 34 Minutes)
2012-06-15 12:56:25 Updated by ITIL User (20 Minutes)

Highlight changes to field

<table>
<thead>
<tr>
<th>Week</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>June 1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>24</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>25</td>
<td>18</td>
<td></td>
<td>22</td>
<td>23</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>July 1</td>
</tr>
</tbody>
</table>
If you hover over the icon within an entry, a popup displays all the value changes. This is the same information that is displayed in the top part of the form.
### Incident History Detail

**Details for INC0000039**

- **Created**: 2012-04-05 17:42:29 by admin
- **Last updated**: 2012-06-15 12:56:26 by itil
- **Update count**: 3 (1 audited)

**2012-04-05 17:42:29** Created by System Administrator (70 Days 11 Hours 46 Minutes 1 Second)

**2012-06-15 12:56:26** Updated by ITIL User (2 Minutes)

Highlight changes to field: **None**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>true</td>
</tr>
<tr>
<td>Approval</td>
<td>Not Yet Requested</td>
</tr>
<tr>
<td>Assignment group</td>
<td>Network</td>
</tr>
<tr>
<td>Caller</td>
<td>Bud Richman</td>
</tr>
<tr>
<td>Category</td>
<td>Network</td>
</tr>
<tr>
<td>Configuration item</td>
<td>MailServerUS</td>
</tr>
<tr>
<td>Additional comments</td>
<td>Routing from San Diego to the Oregon mail server appears to be getting packet lost!</td>
</tr>
<tr>
<td>Contact type</td>
<td>Phone</td>
</tr>
<tr>
<td>Escalation</td>
<td>Normal</td>
</tr>
<tr>
<td>Impact</td>
<td>3 - Low</td>
</tr>
<tr>
<td>Incident state</td>
<td>New</td>
</tr>
<tr>
<td>Knowledge</td>
<td>false</td>
</tr>
<tr>
<td>Location</td>
<td>Salem OR</td>
</tr>
<tr>
<td>Made SLA</td>
<td>false</td>
</tr>
<tr>
<td>Notify</td>
<td>Do Not Notify</td>
</tr>
<tr>
<td>Number</td>
<td>INC0000039</td>
</tr>
<tr>
<td>Opened</td>
<td>2012-04-05 17:41:01</td>
</tr>
<tr>
<td>Opened by</td>
<td>Bud Richman</td>
</tr>
<tr>
<td>Priority</td>
<td>4 - Low</td>
</tr>
<tr>
<td>Severity</td>
<td>3 - Low</td>
</tr>
<tr>
<td>Short description</td>
<td>Routing to Oregon mail server</td>
</tr>
<tr>
<td>SLA due</td>
<td>2012-04-26 17:41:01</td>
</tr>
<tr>
<td>State</td>
<td>New</td>
</tr>
<tr>
<td>Task type</td>
<td>Incident</td>
</tr>
<tr>
<td>Domain</td>
<td>global</td>
</tr>
<tr>
<td>Urgency</td>
<td>3 - Low</td>
</tr>
</tbody>
</table>

**April 2012**

<table>
<thead>
<tr>
<th>Sun</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>April 1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Week**

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>17</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>18</td>
<td>30</td>
<td>May 1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
You can click on the day number to get a view of the changes for that day. You can also click on the week number to the left to get a the week view. You can scroll to and from month to month to see changes.

Timeline of CI changes

You can view a timeline of changes for a CI and its relationships, and proposed changes for the CI. Timelines are available for CIs in the Configuration Item [cmdb_ci] table or a descendant of this table, if auditing is enabled for the tables.

Role required: The ACL for this view is based on the roles defined in the glide.history.role system property, which by default is set to util. Also, the user must have read access to the History Set [sys_history_set] table, which by default is granted to admin.

You can open a timeline when you view the history of a CI. You can specify the time period, time range, and properties that are displayed in the timeline. You can view either what has changed in a particular change set, or view the entire CI to better troubleshoot any issues. You can also export and compare snapshots of the CI at any point in time.

CI changes are represented by bubbles in different shapes and colors along the timeline. The shape of each bubble represents a different type of change and the color of each bubble specifies whether the change is valid or invalid. Click the ? icon to display bubble shape and color definitions, and point to a bubble to display details about the change set.

A change to a relationship is considered valid only if it was applied through change management. If the change was applied via the Proposed Changes framework - then it is valid.

---

**Figure 723: History Timeline view**
Figure 724: Timeline bubbles

Note: Proposed changes that do not have a planned start date are placed at future points of time.

Timeline navigator

Use the handles on both ends of the timeline navigator to extend or to shorten the time period that is shown.

You can scroll to a different period of time by clicking on the bottom part of the timeline navigator and then dragging the navigator to the left or right.

Zoom

By default, the timeline for the last month is shown. Next to the Zoom label above the timeline, you can select another time interval. You can select intervals from a minute to the entire period of data.

If there are many changes of the CI during the time period, the bubbles displayed might get too crowded. You can zoom in or out to spread the bubbles in either method:

- Change the time interval on the timeline. As you shorten the time interval, you zoom in, and as you lengthen the time interval, you zoom out.
- Select the section of the timeline that you want to zoom into.

Property filter

You can filter the bubbles that are displayed. By default, all bubbles are displayed, representing changes to all of the CI's properties. You can limit the view to display only the bubbles in which selected properties have changed and exclude bubbles in which only unselected properties changed.

The Detail and Summary views highlight properties within your filter scope that have changed. The changed properties are highlighted in light blue.

In the Summary view, you can choose to include all the properties of the CI, or only properties that have changed. If you choose to display all properties in the summary view, then changed properties are listed before unchanged properties.
Summary view

The **Summary** view displays snapshots of the CI's represented by each bubble. Each snapshot displays the changes to the CI's fields and relationships according to the change set. It displays old and new values before and after the change, and any relationships that were added or deleted.

Use the > and < buttons on both sides of the snapshot display to scroll through the next and previous change set records in a chronological order.

Detail view

The **Detail** view displays snapshots of the CI that correspond with the bubbles. Each snapshot includes the fields that are within the property filter scope, displaying the properties that have changed with a light blue background. Click on a bubble to display its corresponding snapshot of the CI. The data that is displayed is read-only.

Use the > and < buttons on both sides to scroll through the next and previous change set records in a chronological order.

**View timeline of changes to related records**

On the timeline of changes for a CI view, you can also view a timeline of changes to the CI's related records.

The CI must be in the Configuration Item [cmdb_ci] table or a descendant of this table. Auditing must be enabled for the table containing the CI.

Role required: admin

1. Open the timeline for the CI.
2. Click the Related Records icon and select related records from the List of Related Records to view. Click the Related Records icon again to display the related records timeline.

The timeline of changes to the CI's related records is displayed above the CI's timeline. If you uncheck all related records, the related records timeline is hidden.

Hover over a change bubble on the related records timeline to display details about the change, such as date and number of changed properties. As you change the time interval in focus, or zoom in or out, it affects both the CI timeline and the related records timeline simultaneously.

**Export a snapshot of a CI**

You can export a snapshot of a CI from its timeline.

The CI must be in the Configuration Item [cmdb_ci] table or a descendant of this table. Auditing must be enabled for the table containing the CI.

Role required: admin

You can export a snapshot of the CI to an XML, PDF (Portal), or PDF (Landscape) format.

1. Open the timeline for the CI.
2. Select the bubble representing the time for which you want to export a snapshot of the CI.
3. Click the export icon ( ).
4. Select the file format to use for the export. You can download the file to your system for viewing.

**Compare CI snapshots**

You can compare the properties and relationships of a CI at two different points in its timeline.
The CI must be in the Configuration Item [cmdb_ci] table or a descendant of this table. Auditing must be enabled for the table containing the CI.

Role required: admin
1. Open the timeline for the CI.
2. Click **Compare**.
3. Select a **Start** date and an **End** date.
4. Click **Compare**.

**Restore deleted records**

In some cases, administrators can restore deleted records and references to those records.

For example, if a user inadvertently deletes a user record that was referenced in the **Caller ID** field on several incident records, you may be able to restore the user record as well as the incident field values. You may also be able to restore records that were deleted as a result of a cascade delete action.

There are several methods for restoring deleted records:

- Restore **data records without references** on tables that audit deletions.
- Restore data records and references on tables that audit deletions (requires the Restore Deleted Records plugin).
- Restore configuration records with the app creator.

**Restore deleted records**

You can restore data records without references on tables that audit deletions.

Role required: admin
1. Navigate to **System Definition > Deleted Records**.
2. Select the check boxes beside the records you want to restore.
3. In the **Actions** choice list below the list, select **Undelete Records**.

![Actions on selected rows](image)

**Restore data records with deletion audits**

You can restore deleted data records on tables that audit deletions.

You can restore deleted data records on tables that audit deletions. Limitations for restoring data records include:

- Record deletions are not tracked for tables with the `no_audit_delete=true` dictionary attribute.
• **Attachments** are not restored when the record is restored.
• Deletions from tables with a sys prefix are not audited by default. You can use the app creator to restore configuration records, such as Business Rules [sys_script]. You can also configure specific system tables to audit deletions.
• References are restored only if the reference field is on an audited table and the Restore Deleted Records plugin is activated.
• References that use an Image field type are not restored.

**Configure a system table for a deletion audit**

Deletions from tables with a sys prefix are not audited by default. To track deletions from these tables, add the table name to the glide.ui.audit_deleted_tables property.

Role required: admin

Enabling the Restore Deleted Records plugin adds several default values to this property.

1. Navigate to System Properties > UI Properties.
2. Locate the List of system tables (beginning with "sys", comma separated) that will have the delete audited property.
3. Add or remove table names. Table names should be separated by commas, without any spaces.
4. Click Save.

**Note:** For more information about auditing, see Understanding the sys audit Table.

**Restore a deleted configuration record**

You can use the app creator to restore deleted configuration records with related records and references.

Role required: admin

Limitations for restoring configuration records include:

• You cannot restore a record if its parent record is deleted. For example, you cannot restore a field on a deleted table. You must restore the parent record first.
• When a record is deleted, the application file stores a reference to the current record version. You can only restore records when this reference and version record exist.

To restore a configuration record:

1. Navigate to System Applications > Deleted Application Files.
2. Locate the configuration record you want to restore. For example:
   • For an application, filter on Table name is Custom Application and look for the application name.
   • For a table, filter on Table name is Table and look for the table name.
   • For a field, filter on Table name is Dictionary Entry and look for the field name.
3. Open the record.
4. Review the related records and references that will be restored with the selected record. These related lists display any applicable records.
Geneva  ServiceNow  ServiceNow Platform

• Deleted Files - Descendants: configuration records that were deleted with the application, table, or field. For example, restoring a table also restores all the fields and labels that were deleted with the table.
• Deleted Files - References: configuration records that referenced the record and were deleted because of a cascade delete rule.
• Deleted Data: data records on audited tables that referenced the record and were deleted because of a cascade delete rule.
• Cleared References: reference field values that were cleared in data and configuration records.

5. Click Restore File.
6. In the confirmation dialog, click OK.
   The record and all restorable related records and references are restored.

Domain separation

Domain separation (also known as multi-tenancy) is a way to separate data into logically-defined domains. Domain separation is best for those customers who:
• Need to enforce absolute data segregation between business entities (data separation).
• Customize business process definitions and user interfaces for each domain (delegated administration).
• Maintain some global processes and global reporting in a single instance.
• Separate data between customers or sub-organizations.
• Have minor or moderate process differences only among customers or sub-organizations.

Alternatives to domain separation include:
• Before business rules
• Access control list rules (ACLs)
• Filters
• Security on related record
• Custom views
• Form layouts
• Notifications
• UI action conditions
• Advanced reference qualifiers

Warning: Before activating domain separation, consult your representative to verify that it is suitable for your environment. Domain separation adds a level of administration overhead. Although it can be disabled, it cannot be removed from an instance.

Data separation

Members of a domain only see the data contained within their domain or the child domains that are lower in the domain hierarchy. By default, all users and all records are members of the global domain unless an administrator assigns them to a particular domain. Once you assign a user or a record to a domain, the instance compares the user's domain to the record's domain to determine whether the user can view the record. For example, consider the following domain hierarchy:
Geneva    ServiceNow    ServiceNow Platform

Figure 725: Sample domain hierarchy

Domain: Global

Domain: Database

- MIL User
- Beth Anglin
- Fred Luddy

Domain: Database

- Atlanta
  - Bow Ruggeri

Domain: Database

- San Diego
  - Don Goodliffe

Domain: NY DB

- David Loo

Records in the global domain are visible to all users.

Users in a parent domain can see and manage records in child domains.

Users in a child domain cannot see or manage records in a parent domain.
In this domain hierarchy:

- Bow Ruggeri can see any records in the Database Atlanta or the global domain.
- Don Goodliffe can see any records in the Database San Diego or the global domain.
- David Loo can see any records in the NY DB or the global domain.
- Fred Luddy, ITIL User, Beth Anglin can see any records in the Database, Database Atlanta, Database San Diego, NY DB, or the global domain.

Users in the global domain can see all records, regardless of the record's domain settings. If a user is a member of another domain, then there is no single visibility setting that allows users to see across domains or allows users to see records at a higher level in the hierarchy.

**Note:** Guest users must be part of the global domain.

### Exceptions to data separation

In general, data defined at a higher level in the domain hierarchy is not visible at lower levels in the hierarchy. However, the following records behave like policies:

- Form sections
- Options in a choice list

When defined at a higher level in the hierarchy, these records are visible in child domains.

### Request domain separation

All domain support features are activated with one plugin called Domain Support - Domain Extensions. Administrator can request activation of this plugin.

Role required: admin

The plugin replaces these plugins from previous releases:

- Domain Support (version 2.0)
- Domain Support - Common
- Domain Support - MSP Extensions

1. In the HI Service Portal, click **Service Catalog** > **Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
<tr>
<td>Specify the date and time you would like this plugin to be enabled</td>
<td>Date and time must be at least 2 business days from the current time.</td>
</tr>
</tbody>
</table>

**Note:** Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.
3. Click **Submit**.

### Domain assignment

By default, domain separation adds a domain field to the Task [task] and Configuration Item [cmdb_ci] tables and their extensions.

You can also extend domain separation to any new tables you create by adding a **sys_domain** field to the table's dictionary definition. By default, the system only domain separates platform and baseline application tables where appropriate.

**Warning:** ServiceNow does not recommend domain separating platform tables (any table with the sys_ prefix such as the Dictionary Entry [sys_dictionary] and Dictionary Entry Override [sys_dictionary_override] tables) because it can produce unexpected results.

The value of the **sys_domain** field contains the domain assigned to the record by any of the following:

- Company to which the user belongs
- Business rule when creating record
- Module used when creating record
- Form template used when creating record
- Domain of the parent record
- Domain assigned to User record
- Domain of the user who creates it

**Domain separation restrictions:**

The system prevents the following tables from being domain separated:

- **Access Control** [sys_security_acl]
- **Script Include** [sys_script_include]
- **System Property** [sys_properties]
- **Security Black/Whitelist Entities** [sys_security_restricted_list]
- **Dictionary Entry** [sys_dictionary]
- **Dictionary Entry Override** [sys_dictionary_override]

### User-based assignment

Administrators can quickly assign users to a domain by assigning them to a company. After users are assigned to a domain, records automatically inherit the user's domain.

Administrators can automatically assign users to domains:

- Based on a company record
- Based on a default domain
For example, assigning Bow Ruggeri to the ACME company automatically assigns him to the ACME domain. Assigning Don Goodliffe to the Initech company automatically assigns him to the Initech domain. Any records they create are automatically added to the appropriate domain.

Figure 726: Adding a user to a domain by setting the company

**Business-rule-based assignment**

Administrators can use a business rule to automatically set a domain value when creating a record. The business rule must set a value in the `sys_domain` field. Administrators must ensure there is a `sys_domain` column available for the record's table.

**Module-based assignment**

Administrators can use the `sysparm_domain` URL parameter to automatically assign new records to a particular domain from a module. Administrators must create a module with an **Argument** value of: `sysparm_domain=sys_ID of domain`.

**Template-based assignment**

Administrators can use a form template to automatically assign new records to a particular domain. Administrators must add the `sys_domain` field to the form and select a domain value. For example, setting the `sys_domain` field to **TOP/ACME domain** automatically assigns all records from this template to the TOP/ACME domain.

**Table-class-based assignment**

By default, related records inherit the domain of the parent record. For example:

- A change task record inherits the domain of the parent change request record.
- A problem record inherits the domain of the parent incident record.
Record-based assignment

If no other domain conditions apply, a record automatically inherits the domain of the user who creates it.

Domain visibility

Domain visibility determines whether users from one domain can access records from another domain. For example, if Don Goodliffe is in the Database domain, and Bow Ruggeri is in the Network domain, and no incidents are in the global domain, then Don Goodliffe cannot access Bow Ruggeri’s incidents since data separation prevents this.

Note: While visibility is one method to allow users to access records, it is recommended that you use **Contains** for more robust control.

Figure 727: A sample set of domain separated incident records

Figure 728: Bow Ruggeri’s incident list
You can add the Database domain as a Visibility Domain to the Bow Ruggeri's user record (Visibility Domains is a related list on the user record). Then, Bow Ruggeri can access Don Goodliffe's incidents since he now has visibility to the Database domain. If you remove the visibility domain, then Bow Ruggeri can no longer access incidents in the Database domain.

Users can also inherit visibility domains based on their group membership if you set the domain table to the Group [sys_user_group] table. For example, as a member of the Database group, Don Goodliffe also automatically gains the Database domain as a visibility domain. Group membership grants visibility to any matching domain name.
Contains domains

Normally parent-child relationships define the domain hierarchy. A **Contains** domain allows you to relate domains on an as-needed basis, independent of parent-child relationships.

However, contains domains only grant visibility to domain data. Processes remain unaffected by contains relationships.

**Note:** Visibility controls what a particular user can see, while **Contains** controls what an entire domain of users can see.
Contains domains versus visibility domains

Contains domains and visibility domains differ in several respects.

A **contains** domain:

- Is a many-to-many, domain-to-domain relationship.
- Is hierarchical. When a domain is selected, you can see the data from that domain and its children.
- Is controlled by the selection in the domain picker.

A **visibility** domain:

- Is a user-to-domain relationship and is explicitly granted.
- Is not hierarchical.
- Is not controlled by the selection in the domain picker. Once the user is granted access to a visibility domain, they always see data in that domain and its children.

For example, there is a user who has access to domain A (the user's home domain) and is granted visibility to domains B and C. The user selects domain A in the domain picker. In this case, the user has access to domains A, B, and C. If the user changes the domain picker to domain B, B and C are visible. C is still visible because the user still has visibility to it. A is not visible, because it is not selected in the domain picker and it is not a visibility domain.

Using visibility domains excessively is not recommended.

Domain scope

Domain scope defines what users can and cannot have access to.

Every user has two domain scopes when establishing a session in a domain separated instance.

- Session scope is set upon session establishment to the domain listed in the user's user record. Users can manually change their session domain scope from the domain picker.
- Record scope uses the domain of the record and is active when viewing the form of any record.

By default, the record scope takes precedence over the session scope so that fulfillers in higher level domains adhere to each record's data and process constraints. However, these fulfillers can choose to expand or collapse the domain scope to show or hide data from other domains. For example, a user in the MSP domain also has visibility into child domains such as the ACME domain. When looking at an incident record from the ACME domain, the user can choose to expand the domain scope to show values from the MSP domain or collapse the domain scope to only show record values that match the record's ACME domain.

| Note: | Users always have access to data from domains that have been explicitly granted to them by domain visibility. |

Users with the domain expand scope user role can select the domain scope from the **Toggle Domain Scope** UI action on the form. When record scope is in effect, click the UI action to expand to session scope and display all data available based to the user's domain and child domains. When session scope is in effect, click the UI action to collapse to record scope and display only data that matches the current record's domain.

| Note: | A record will not display the UI action to toggle the domain scope if the record is in the global domain or if the user's domain matches the record's domain. |
Record value selection from other domains

Users who can see multiple domains have the option to select record values from a domain that is different than the record's domain.

For example, service desk agents working for a managed service provider might want to assign certain incidents to themselves to resolve issues on behalf of their customers. When they do this, the incident Assigned to field might contain a user from the MSP domain, even though the incident record itself is associated with a child domain such as ACME.

Selecting a record value from another domain does not change the record's domain. The record retains its original domain. When a user views a record with values from multiple domains, the user's domain visibility determines what they see.

<table>
<thead>
<tr>
<th>When these conditions are met</th>
<th>The user has access to these UI elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user has access to the domain of the current record referenced in a field.</td>
<td>The user can:</td>
</tr>
<tr>
<td></td>
<td>• See reference field display value. For example, sees the user name in the Assigned to field.</td>
</tr>
<tr>
<td></td>
<td>• See the related record from reference icon. For example, sees the user record for the user in the Assigned to field.</td>
</tr>
<tr>
<td></td>
<td>• Select values from any visible domain. For example, can select users from either the MSP and ACME domains.</td>
</tr>
<tr>
<td>The user does not have access to the domain of the current record referenced in a field.</td>
<td>The user can:</td>
</tr>
<tr>
<td></td>
<td>• See the reference field display value. For example, sees the user name in the Assigned to field.</td>
</tr>
<tr>
<td></td>
<td>• Only select values from the record's domain. For example, can only select users from the ACME domain.</td>
</tr>
</tbody>
</table>

Delegated administration

Delegated administration allows administrators to set domain-specific policies.

The policies set lower in the domain hierarchy override policies set higher in the domain hierarchy. While in a domain, administrators can set domain-specific versions of these global policies and settings:

- Client scripts
- System policies
- Application and module names
- Application roles
- Module filters
**Warning:** All users with the admin role have special access to all system features, functions, and data because administrators can override ACL rules and pass all role checks. Grant this privilege carefully.

When users have the **admin** role, then all policies in the instance are available to them regardless of the assigned domain. They can enter a specific domain, and then only policies in that domain or higher are visible and processed during a relevant transaction. When an administrator modifies a policy that is in a higher domain or the global domain, the system automatically creates a new record for that administrator's current domain. It does not modify the original policy, application, or module record. This new record overrides the original.

**Note:** To make changes to a policy in a lower-level domain, go into that domain and modify the policy. This approach creates the new policy record in your domain that overrides the original, higher-level policy record.

Do not make changes on the higher-level policy and then change the **Domain** field on that policy. This approach does not create a new policy record in your lower-level domain, nor does it keep the policy record for the higher-level domain.

The **sys_overrides** field indicates that a policy, application, or module at a lower level in the hierarchy overrides a record at a higher level. The system automatically sets this field when an administrator attempts to modify a policy, application, or module that belongs to another domain higher in the hierarchy. Again, rather than actually changing the higher level record, the attempted update is changed into an insert, and the **sys_overrides** field is set to indicate the higher level policy, application, or module that is being overridden. Later when the records for a relevant transaction are loaded, the overriding domain-specific policy, application, or module is used instead of the original.

**Domains for delegated administration**

By default, delegated administration always uses the record's domain to determine what policies to apply. The record's domain takes precedence over the user's domain. If there are no policies in the record's domain, delegated administration checks for policies in the next highest level of the domain hierarchy. The search for domain policies continues up the domain hierarchy until reaching the global domain. If there are no domain policies lower in the domain hierarchy, delegated administration uses the policies for the global domain.

For example, Fred Luddy is a user in the Database domain who can see records in the Database: Atlanta, Database: San Diego, and NY DB child domains. When he opens a record in the Database: San Diego domain, delegated administration first checks for policies in the Database: San Diego domain. If there are no policies at this level of the domain hierarchy, delegated administration checks for policies from the Database domain. If there are no policies in the Database domain, delegated administration uses the global domain policies as there are no other domains higher in the domain hierarchy.

**Example delegated administration with domain specific applications and modules**

The following example illustrates delegated administration with domain-specific applications and modules. As the administrator of the Database domain, David Loo decides to customize the Configuration application. To start with, David reviews the modules available in the Configuration application module.
Figure 732: Starting view of the Configuration application

David decides to rename the Configuration application to CMDB and to allow the inventory_admin role to see the application.
<table>
<thead>
<tr>
<th>Title</th>
<th>Active</th>
<th>Order</th>
<th>Roles</th>
<th>Name</th>
<th>Domain</th>
<th>Overrides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset Management</td>
<td>true</td>
<td>900</td>
<td>admin</td>
<td>asset</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>BSM Map</td>
<td>true</td>
<td></td>
<td>itil</td>
<td>bsmap</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>true</td>
<td>400</td>
<td>itil</td>
<td>change_management</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>CMDB</td>
<td>true</td>
<td>600</td>
<td>inventory_admin</td>
<td>configuration_management</td>
<td>Database</td>
<td>configuration_management</td>
</tr>
<tr>
<td>Content Management</td>
<td>true</td>
<td></td>
<td>content_admin</td>
<td>cms</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Contract Management</td>
<td>true</td>
<td>1,000</td>
<td>contract_manager</td>
<td>asset_management</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Domain Admin</td>
<td>true</td>
<td></td>
<td>domain_admin</td>
<td>domain_admin</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>EOC</td>
<td>true</td>
<td></td>
<td>admin</td>
<td>ecc</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Homepage Admin</td>
<td>true</td>
<td></td>
<td>admin</td>
<td>home</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Incident</td>
<td>true</td>
<td>200</td>
<td>itil</td>
<td>incident_management</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Instance Clone</td>
<td>true</td>
<td></td>
<td>clone_admin</td>
<td>instance_clone</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Knowledge Base</td>
<td>true</td>
<td>800</td>
<td>itil</td>
<td>knowledge</td>
<td>km</td>
<td>global</td>
</tr>
<tr>
<td>Metrics</td>
<td>true</td>
<td></td>
<td>itil_admin</td>
<td>metrics</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>MID Server</td>
<td>true</td>
<td></td>
<td>admin</td>
<td>MID</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Organization Management</td>
<td>true</td>
<td>875</td>
<td>asset</td>
<td>organization_management</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>true</td>
<td>300</td>
<td>itil</td>
<td>problem_management</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>Reports</td>
<td>true</td>
<td>1,100</td>
<td>itil</td>
<td>reports</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>SAML 2 Single Sign-on</td>
<td>true</td>
<td></td>
<td>admin</td>
<td>saml_2_single_sign_on</td>
<td>global</td>
<td></td>
</tr>
<tr>
<td>SAML Single Sign-on</td>
<td>true</td>
<td></td>
<td>admin</td>
<td>SAML_Single_Sign-on</td>
<td>global</td>
<td></td>
</tr>
</tbody>
</table>

Figure 733: Sample domain-specific changes to the Configuration application
Next, David decides to change the Incident application by activating the **Open - in "New" State** module and adding a new filter item to show open incidents in the Database category.

![Sample domain-specific changes to the Open - "New" State module](image)

This creates a new module entry in the application rather than overwriting the existing module in the global domain.

**Figure 734: Sample domain-specific changes to the Open - "New" State module**
Figure 735: Domain-specific view of the Incident application
If another administrator from another domain, such as Fred Luddy, logs in and looks at the Configuration application, he see the settings from the global domain.

Figure 736: David Loo's view of applications
Example delegated administration with domain specific policies

The following example illustrates delegated administration with domain-specific policies. In this hierarchy David Loo is in the Database domain and Don Goodliffe is in the Database/Database San Diego domain. To begin, David Loo makes a change to the global assignment policy.

Then Don Goodliffe also makes a change to the same policy. Initially, all assignment rules have a global domain as shown below:
Figure 738: Global domain rules

If David Loo updates the assignment rule for **Database or Software**, the following list appears:

Figure 739: Database domain-specific rules
The following policy changes occur:

- When the policy is chosen and updated, the system detects that David Loo is not at the right level of the hierarchy to change this record. Therefore, the update is changed into an insert, and a new record is created.
- The new policy has the same name (Database or Software).

Notice that this policy is in the Database domain and overrides the policy that previously applied (Database or Software). Notice that there are now two policy entries with the same name. Because this is not desirable, David opens the record and changes the name to something appropriate. After the update, the list appears as follows.

Figure 740: Renamed database or software to database specific policy

This time, the record being updated is at the same level in the domain hierarchy as the user, so the record is updated with a more appropriate name. Here is the resulting rule. Notice that database incidents will now be directly assigned to David.
Figure 741: Database specific policy assignment rule

If a new incident is created in the Database domain or lower in the hierarchy, the new rule is applied. It has overridden the global assignment rule. If a new incident is created in the global domain or any other domain not within the Database domain hierarchy, then the global rule applies.
In the following scenario, Don Goodliffe, in the Database/Database San Diego domain hierarchy, decides that database incidents created in his domain should be assigned to him rather than to David Loo. As an administrator, Don Goodliffe starts out with the following assignment policy:

![Assignment Rules](image)

Notice that this level of the hierarchy starts out with the policy established at the parent level (the Database domain). After changing the **Database Specific Policy**, the list looks like this:

![Assignment Rules](image)

**Figure 742: Don Goodliffe’s starting view of assignment rules**

Notice that this level of the hierarchy starts out with the policy established at the parent level (the Database domain). After changing the **Database Specific Policy**, the list looks like this:
Figure 743: Database San Diego rules override Database specific policy rules

Again, the attempted update is changed automatically to an insert, and the override value is supplied to indicate that the higher-level policy is being overridden. Here is the resulting rule; it shows that database incidents created in the Database San Diego domain will be assigned to Don Goodliffe.
Figure 744: San Diego specific policy

The result of the above customization is:

- A database incident from the Database San Diego domain will be assigned to Don Goodliffe.
- A database incident from the Database hierarchy other than Database San Diego will be assigned to David Loo.
• A database incident from any other domain, including global, will be assigned to the system administrator.

The above customizations all show changes to higher-level policy. However, new policy can also be created at any level of the domain hierarchy.

During a transaction, the current user's domain normally determines the policy to load. For example when a user in the Database domain updates an incident, the Database domain is used for business rules and policies even if the incident record was originally created in the Database San Diego domain. By default, the user's domain supersedes the record's domain.

There is a system setting that can change this behavior. If Using the Current Record's Domain Instead of the Current User's Domain is set to true, then the above behavior is reversed. The domain of the record is used to determine which policy to load, not the domain of the user. For example if a user in the Database domain updates an incident that is in the Database San Diego domain, then the business rules and policy that exist for Database San Diego are executed. The domain of the user still determines the records that are visible to the user, and the domain of the user sets the domain for records that user creates, but is not a factor in determining rules and policies.

Domain query methods

A domain query method allows the instance to efficiently query large numbers of domains.

Part of Domain Support 2.0 is a new query engine designed to perform and scale to tens of thousands of domains. Prior methods, including domain numbering, have had limitations that domain paths resolves. While you have the flexibility to continue using your existing query method, we highly recommend that you switch to domain paths through the new Domain Configuration screen at your earliest convenience.

Domain paths:

A domain path is a series of three-character codes separated by a slash (/) delimiter that uniquely identifies a domain. Each digit in the three-character code consists of one of the following 60 possible characters:

```
!#$&()*+,-./0123456789:;<@ABCDEFGHIJKLMNOPQRSTUVWXYZ\]^`{|~
```

The three-character codes that make up a path are not unique across a domain tree. Rather, the entire path string itself is unique. For example:
Figure 745: An example domain tree

Table 611: Domain tree example

<table>
<thead>
<tr>
<th>Domain name</th>
<th>Parent domain</th>
<th>Domain path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNC</td>
<td>None</td>
<td>!!!/</td>
</tr>
<tr>
<td>SNC/US</td>
<td>SNC</td>
<td>!!!/!!!/</td>
</tr>
<tr>
<td>SNC/EU</td>
<td>SNC</td>
<td>!!!/!!!#/</td>
</tr>
<tr>
<td>SND/RU</td>
<td>SNC</td>
<td>!!!/!!$/</td>
</tr>
<tr>
<td>SNC/US/NY</td>
<td>SNC/US</td>
<td>!!!/!!!#/</td>
</tr>
<tr>
<td>SNC/US/CA</td>
<td>SNC/US</td>
<td>!!!/!!!#/</td>
</tr>
<tr>
<td>SNC/EU/DE</td>
<td>SNC/EU</td>
<td>!!!/!!!#/</td>
</tr>
<tr>
<td>SNC/EU/FR</td>
<td>SNC/EU</td>
<td>!!!/!!!#/</td>
</tr>
</tbody>
</table>

Note: With three-character codes delimited by a single character in a path string of 255 total characters, each node of the domain tree supports up to 216,000 child domains, and the maximum depth of the tree is 63 levels.

Domain separation setup

Setting up domain separation involves activating the plugin, setting options, and assigning users and records to domains.

Activating the Domain Extension plugin enables these features:

- Domain separation is based on the Domain [sys_domain] table.
- Delegated administration lets each domain have separate policy.
- All records are part of the global domain.
• The current user’s domain determines the domain to use when viewing or operating on a record in a different domain.

Domain separation replaces Company Separation. However, if company separation is already active when you activate domain separation, both plugins are active at the same time. You can control the company separation activation status with the glide.db.separation.field property.

Domain support plugins

All domain support features are consolidated into one plugin called Domain Support, version 2.0.

The plugin combines the features of the previous plugins:

• Domain Support (version 1.0)
• Domain Support - Common
• Domain Support - Partitioning (Data separation)
• Domain Support - Delegated Administration
• Domain Number Support

In addition, the Domain Support version 2.0 offers these new features:

• Choice of domain query method
  • Domain Paths (new)
  • Domain Numbers
• All-new Domain Admin UI
• Improved stability and scalability

Activate or deactivate a domain

MSP Extensions allows you to activate or deactivate a domain, which cascades the activation status to companies in the domain.

Role required: admin

When you activate a company record, MSP Extensions automatically activates the company's associated domain. For example, if you activate the ACME company, then you also activate the TOP/ACME domain.

1. To deactivate a domain, navigate to the domain record.
2. Clear the Active check box.
3. Click Update.

**Warning:** Do not delete domains. Deactivate domains that you no longer need instead of deleting them.

Grant visibility domains to an individual user

How to grant visibility domains to an individual user.

Role required: admin

1. Navigate to **User Administration** > **User**.
2. Select the user you want to provide with visibility domains.
3. Add the **Visibility domains** related list to the form.
4. From the Visibility Domains related list, click **Edit**.
5. Select the domains whose records you want the user to see.
6. Click **Save**, and then click **Update**.

![Visibility domain embedded list](image)

The Visibility domain embedded list contains the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Domain that is visible to the group or user.</td>
</tr>
<tr>
<td>Inherited</td>
<td>Domain is inherited from domain visibility or a parent domain.</td>
</tr>
<tr>
<td>Granted By</td>
<td>Name of the group that granted domain visibility.</td>
</tr>
<tr>
<td>Parent visibility</td>
<td>Name of the parent domain and used for grouping records. If the parent record is deleted, then all records with the same parent are deleted as well.</td>
</tr>
</tbody>
</table>

**Select the domain query method**

Converting from domain numbers to domain paths requires a one-time background conversion process.

Role required: admin

This process can take several hours depending on the number of domains to be converted. Consider scheduling the conversion during off-peak hours to minimize the performance impact on users.

1. Navigate to **Domain Admin > Configuration**.
2. From **Query Method**, select either **Enable Domain Paths** or **More > Enable Domain Numbers**.
3. Click **OK** to confirm the selection.

**Note:** The conversion duration depends upon the current number of domains in the instance. The more domains there are, the longer the conversion takes to add path or number references to the domains.

**Configuration module**

The Domain Configuration module is the central location where an administrator enables, configures, and views the status of domain separation support.
Figure 746: Domains Configuration

The Configuration module consists of these fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Table</td>
<td>Select the table containing domain names for domain separation. You can select any existing table. By default, the system uses the Domain [domain] table. <strong>Warning:</strong> No domain can have the name global. Verify that the name global is not used in any of the domain names in the table before saving the domain configuration.</td>
</tr>
<tr>
<td>Domain Validation</td>
<td>Lists any current validation issues. Click the More link to see validation troubleshooting options.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Domain Progress Workers</td>
<td>Lists any currently running conversion or validation processes.</td>
</tr>
<tr>
<td>Domain Alerts</td>
<td>Lists any information, warning, or error messages relating to domain separation. You can also find</td>
</tr>
<tr>
<td></td>
<td>this information in the instance log.</td>
</tr>
<tr>
<td>Enable Domain Separation</td>
<td>Select whether to enable or disable domain separation. By default, domain separation is enabled.</td>
</tr>
<tr>
<td></td>
<td>You can manually disable domain separation with this setting. However, the sys_domain column will</td>
</tr>
<tr>
<td></td>
<td>still be present on any table it was added to. This option maps to the system property glide.sys.domain.partitioning.</td>
</tr>
<tr>
<td>Enable Delegated Administration</td>
<td>Select whether to enable or disable delegated administration. Activating the plugin automatically</td>
</tr>
<tr>
<td></td>
<td>enables delegated administration. You can manually disable delegated administration with this setting.</td>
</tr>
<tr>
<td></td>
<td>This option maps to the system property glide.sys.domain.delegated_administration.</td>
</tr>
<tr>
<td>Enable verbose domain logging</td>
<td>Select whether to enable additional debugging information for domain separation. This option maps to</td>
</tr>
<tr>
<td></td>
<td>the system property glide.sys.domain.verbose.</td>
</tr>
</tbody>
</table>

### Domain separation options

You can set the following domain separation configuration options by navigating to **Domain Admin > Configuration**.

- Select the **domain query method**.
- Provide a list of roles that ignore the domain of the current record.
- **Enable domain log and debug messages**.
- **Add domains to a visibility domains list** on page 2609.
- Create a contains relationships between domains.
- Create domain-specific personalizations.
- Prevent a user from seeing the complete domain list.
- Display **domain selection menus**.

### Add a domain field to a table

Administrators can domain separate custom tables by adding a sys_domain field to it.

Role required: admin

1. Navigate to the table’s list view. For example, type `<table name>.list` in the navigation filter.
2. Right-click the list header and select **Configure > List Layout**.
3. In the **Create new field** section, enter `sys_domain` as the Name and Domain ID as the Type.
4. Click **Add**.
5. Click **Save**.
**Note:** Any other means of creating a field adds a `u_` prefix to the column name. For domain separation to work the column name must be `sys_domain` without any `u_` prefix.

---

**Add domains to a visibility domains list**

Adding a visibility domain allows a user or group to see and potentially edit records from another domain regardless of the user or group's normal domain membership.

Role required: admin

Assigning visibility domains to all members of a group is preferred over granting them to individual users.

**Note:** Adding a visibility domain does not change a table or record's access control rule requirements.

1. Navigate to the domain table.
2. Select the group you want to provide with visibility domains.
3. Add the **Visibility domains** related list to the form.
4. From the **Visibility domains** related list, click **Edit**.
5. Select the domain records you want the group or domain to see.
6. Click **Save**, and then click **Update**.

---

**Create contains relationships between domains**

Creating a contains relationship between domains changes the domain hierarchy.

Role required: admin

Domains in a contains relationship inherit the visibility settings of the containing domain. The containing domain allows users to see data in the contained domain as well as any of its children. Processes are unaffected by a contains relationship.

1. Navigate to the domain table.
2. Select the domain record that will be the parent (container) domain of the new contains relationship.
3. **Toggle the domain scope** to switch between the session scope and record scope, if necessary.
4. From the Contains Domains related list, click **Edit**.
5. Select the domain records that is the child (contained) domains of the contains relationship. Only child domains appear by default when the domain picker is set to Global. Toggle the domain scope to see all domains in slushbucket.
6. Click **Save**, and then click **Update**.

---

© 2017 ServiceNow. All rights reserved.
Create a domain-specific choice list
Administrators can configure choice lists to contain entries specific to a particular domain.

Role required: admin

1. Select the domain from domain picker where the choice should be added.
2. Right-click the field and select **Configure Choices**.
3. Update or add choices.
4. Push changes through the normal change process such as update sets.

**Note:** Administrators should ensure that choices are unique across domains to prevent administrative confusion in the global domain.

If an administrator adds a new choice from the global domain, then users from domains lower in the hierarchy see the new choice at the end of their current choice lists. If the new choice is not active at the global level, then it is available to the domain users via **Configure Choices** but does not show as an active choice.

**Domain selection menus**
The instance offers domain selection via two menu formats.

The available menu formats are:

- **Simple domain picker:** provides a simple drop-down list of available domains.
- **Domain reference picker:** enables a reference field that offers filtering and an auto-complete, type-ahead, entry feature. This format is suitable for longer lists.

The placement of these pickers and the procedure to show or hide them differ depending on the user interface version.

Enable the simple domain picker on UI11 and UI15
The simple domain picker is a drop-down list of available domains and does not support the auto-complete, type ahead feature.

Role required: admin

On UI11 and UI15, you must enable a UI macro. On UI16, you can use the **General** tab on the **System Settings page** of the banner frame.

1. Navigate to **System UI > UI Macros**.
2. Select the **domain_select** macro.
3. Select the **Active** check box and click **Update**.
4. Refresh the browser.
5. Click the gear icon in the banner frame (UI15) or the Domain choice list (UI11).

![Domain Picker](image)

**Figure 747: The simple domain picker for UI15**

Enable the domain reference picker for UI11 and UI15

Use the domain reference picker with auto-complete for long lists that require excessive scrolling.

Role required: admin

On UI11 and UI15, you must enable a UI macro. On UI16, use the General tab on the System Settings page of the banner frame to enable only the simple domain picker. Starting with Geneva patch 4, you can enable a property to enable this domain reference picker.

1. Navigate to System UI > UI Macros.
2. Select the `domain_reference_picker` macro.
3. Select the `Active` check box, and click `Update`.

   **Note:** The `domain_select` macro must be set to `false`.

4. Refresh the browser.
5. Click the gear icon in the banner frame to show the `Domain` field (UI15). The `Domain` field appears by default on the banner in UI11.

![Figure 748: The domain reference picker for UI15](image)

6. Click the magnifier icon in this field to open a list of domains that can be filtered for easy selection.
7. To use the auto-complete feature, type a letter in the field. The menu displays all domains beginning with that letter.
Restrict access to domain selection by role (UI15 and UI11)

You can include a comma-separated list of roles in domain-related UI macros to restrict access to domain selection by user role.

Role required: admin

1. Navigate to System UI > UI Macros.
2. Select any of the four domain-related UI macros: two named domain_select and two named domain_reference_picker.

There are two of each type two support different UIs. The UI macros with the Media Type field empty are for UI11, and the other two are for UI15.

3. Locate one of the following lines of code:
   - domain_select:

```javascript
var ds = new DomainSelectControl("gsft_domain","itil","Domain","Domain");
```
• **domain_reference_picker**:

```javascript
var dp = new
DomainPicker("${jvar_id}" ,"itil", "DomainReference", "Domain");
```

4. Replace the parameter itil with a comma-separated list of roles to control who sees the domain selection menu.

**View domain relationships**

The domain map offers domain administrators a read-only representation of the active domains on the instance and how they relate to each other.

Role required: admin

All domain maps must have one domain set as the primary domain. In addition, each domain in the domain map must meet these criteria:

- The **Parent** field must be filled in (the primary domain is the only exception to this).
- The **Active** check box must be selected.

The domain map does not draw domain relationships for domains that fail to meet the mapping criteria.

1. Navigate to **Domain Admin > Domain Map**.
2. Click the plus (+) or minus (-) icons on the domain headers to show or hide sub domains.

**Select a primary domain**

The primary domain indicates the top level domain in the domain map.

Role required: admin

The primary domain cannot have a parent domain and must have at least one child domain. There can only be one primary domain at a time. If you select another domain as the primary domain, it overrides the previous primary domain.

1. Navigate to **Domain Admin > Domains**.
2. Select the domain you want to be the primary domain. For example, TOP.
3. Select the **Primary** check box.
4. Click **Update**.
Domain

Name: TOP
Type: MSP
Primary: 

Description:
Top level, process flows down from here. Overrides from global process are done here.

Default:
Update Delete

Companies

New Go to Name

Domain = TOP

Name Street City Zip / Postal code Phone Updated

Contains Domains

New Edit... Go to Contains

Domain = TOP

Contains

Contained By

New Edit... Go to Domain

Contains = TOP

Domain

TOP/MSP

Actions on selected rows...
Create a domain
You can create a new domain by creating a new record in the [domain] table.

Role required: admin

When creating a new domain, keep the following in mind:

- Only one domain can be the default domain.
- Only one domain can be the primary domain.

1. Navigate to Domain Admin > Domains.
2. Click New.
3. Fill in the necessary fields (see table).
4. Click Submit.

![Domain form fields](image)

Table 612: Domain form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the domain.</td>
</tr>
<tr>
<td>Type</td>
<td>Select a domain type that describes the domain. By default the domain types are Vendor, Customer, and MSP. You can also add your own choices.</td>
</tr>
<tr>
<td>Primary</td>
<td>Select the check box if this domain is to be the top level domain in the hierarchy. The top level domain only has child domains and no parent domains.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Parent</td>
<td>Select the name of the domain higher in the hierarchy that contains this domain. This field must have a value for the domain to appear in the domain map.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to make the domain available for use. You must select this option for this domain to appear in the domain map.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description for the domain.</td>
</tr>
</tbody>
</table>

Each domain record can also have several related records:

- Companies
- Contains Domains
- Contained By

To change the domain hierarchy, go to the Contains Domains related list and select the domain records that will be the child (contained) domains of the contains relationship.

**Change domain visibility**

By default, when a user in the global domain views a table containing a `sys_overrides` column, the user sees records from only the global domain.

Role required: admin

1. Change the glide.sys.restrict_global_domain_processes property to `true`.
2. To view records from all domains, click **Expand Domain Scope** under Related Links.
3. To return to viewing records from the global domain only, click **Collapse Domain Scope**.

**Deactivate a domain**

Brief introduction to the topic that appears on overview pages and in link previews.

Role required: admin

⚠️ **Warning:** Do not delete domains. Deactivate domains that you no longer need instead of deleting them.

1. Open the domain record.
2. Deselect the **Active** option.

3. Save the record.
Domains and associated companies
Domain separation allows you to cascade changes you make to a company record to the domain and other records associated to the company.

By default, the system automatically assigns users to the same domain as their company. For example, all users of the ACME company automatically become members of the TOP/ACME domain.

Note: Users with the admin role have the ability to change their own user records and therefore can change domains. Managed Service Providers may want to either disable delegated administration or set up an approval process to verify that the user needs the admin role.

When you change a company’s domain, the instance automatically changes the domain of the following associated records to match the company’s new domain.

• Locations
• Departments
• Groups
• Users

When you change a company’s domain, the instance automatically changes the domain of the following associated records to match the company’s new domain.

Note: The instance does not automatically change the domain of any record where you have selected the Managed domain checkbox.

Domain deactivation and associated companies:
When you deactivate a domain, the instance also automatically completes the following actions.

• Deactivates all companies in the domain.
• Prevents all users assigned to the inactive company from logging in.

Note: When a user from an inactive company attempts to log in, the user sees an error message.

For example, if you deactivate the ACME domain from the sample data, the instance also deactivates the ACME company, and the three sample users are locked out.

Figure 750: Login error message example
Default domain scope
Administrators have access to properties and user preferences that control domain scope.

Properties

New activations of domain separation automatically restrict domain scope to the record’s domain for all processes. When the property is set to **true**, the record's domain is used for all process when the user is viewing any record from the form view. If the form has tabs, each tab is considered to have its own domain based on the record.

<table>
<thead>
<tr>
<th>Property</th>
<th>Details</th>
</tr>
</thead>
</table>
| glide.sys.domain.use_record_domain_for_processes | Restricts domain scope to the record's domain for all processes. This property does not apply to business rules. Business rules are always processed from the domain record.  
  - Type: true | false  
  - Default value: true  
  - Location: System Property [sys_properties] table |
| glide.sys.domain.use_record_domain_for_data   | Specifies what domain sets the view of data on a record. Options are to use the record's domain (true) or the user's domain (false). Instances on Fuji should always have this property value set to true.  
  - Type: true | false  
  - Default value: false |

When the glide.sys.domain.use_record_domain_for_processes property is set to **true**, the following properties are not used, regardless of their setting:

- glide.sys.domain.use_record_domain
- glide.sys.domain.use_record_domain_for_client_scripts
- glide.sys.domain.domain_change_notify
- glide.sys.domain.no_change_roles

User preferences

In addition, user administrators can set the following user preference globally or on a per-user basis:
## Table 614: Domain scope user preferences

<table>
<thead>
<tr>
<th>Preference</th>
<th>Category</th>
<th>Updated By</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.domain.session_scope</td>
<td>Domain</td>
<td>Admin Only</td>
<td>When true, sets the default scope to the user's session domain rather than the record's domain. When false, the default scope is the record's domain. Users with the domain_expand_scope user role can still change the domain scope as needed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Type</strong>: true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Default value</strong>: false</td>
</tr>
<tr>
<td>glide.domain.session_scope_notification</td>
<td></td>
<td>Admin Only</td>
<td>When true, displays a visual cue that record values include an expanded domain scope. When false, the notification is hidden.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Type</strong>: true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Default value</strong>: true</td>
</tr>
</tbody>
</table>

### Add a user and a record to a domain

By default, all users and records are part of the global domain and are therefore accessible to all users.

Role required: admin

To use data separation and delegated administration, you must first assign users and their relevant records to one or more domains.

1. Personalize the list or form to add the Domain field.
2. Set the Domain field for the user or record.

For example, assign Bow Ruggeri to the Network domain. Assign Don Goodliffe to the Database domain. Add the Network and Database domains to relevant records such as incidents, configuration items, requests.
Make a domain the default
The default domain is the domain to which the system automatically assigns task and user records that are not already assigned to a domain.

Role required: admin

1. Navigate to **Domain Admin > Domains.**
2. Open the domain you want to be the default domain. For example, Default.
3. Configure the form layout to add the **Default** field.
4. Select the **Default** check box.
5. Click **Update.**
Tasks and users without a domain are placed in this domain when created/updated. This behavior can be overridden by:
1) Unchecking the Default field on this record, or
2) Checking the Default field on another Domain record.
If no domain is set as Default, Tasks and Users with no domain are placed in the global domain.

<table>
<thead>
<tr>
<th>Name:</th>
<th>Default:</th>
<th>Parent:</th>
<th>MSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>MSP</td>
<td>Active:</td>
<td></td>
</tr>
<tr>
<td>Primary:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Companies

- **Go to**: Name
- **New**
- **Domain = Default**
  - **Contains Domains**
    - **New**
    - **Contains**
  - **Domain**
- **Contains By**
  - **New**
  - **Domain**
Manually manage the domain for particular records

By default, the system automatically assigns a domain based on the user's company record. In some cases, however, domain administrators want to manually manage which domain a particular record belongs to.

Role required: admin

The **Managed domain** field allows domain administrators to manually select a domain for the user, group, department, location, or CI record, rather than using the domain assigned automatically from the company record. The **Managed domain** field is available on these record types.

- User records
- Group records
- Department records
- Location records
- CI records

1. Navigate to the record you want to manually manage.
2. Select the **Managed** domain check box.
3. From the **Domain** field, select the domain for the record.
4. Click **Update**.
Clearing the Managed domain check box hides Domain field and the record uses the domain value from the record's company.

Provide a list of roles that ignore the domain of the current record

By default, all roles use the domain of the current record when Use the domain of the record being viewed instead of the user's own property is true.

Role required: admin

You can provide a list of roles that ignore this property and always use the user's domain rather than the record's domain. You may want certain roles such as administrators to always work from their own domain rather than use the domain of the record they are viewing.

1. Navigate to Domain Admin > Configuration.
2. For List of roles (comma-separated) that will not trigger the automatic change of domain to the domain of the record that is being viewed, enter a comma-separated list of roles that ignore automatic domain change behaviors.
3. Click Save.

Use a custom table for the domain table

You can use a custom table as the domain table if the custom table contains a reference field column called parent that refers back to the custom table.
Role required: admin

1. Create a custom table to store the domain information. For example:

<table>
<thead>
<tr>
<th>Table</th>
<th>Column name</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_organization</td>
<td>u_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>u_organization</td>
<td>u_description</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>u_organization</td>
<td>u_location</td>
<td>reference</td>
<td>cmn_location</td>
</tr>
</tbody>
</table>

2. Create a reference field within the custom table that refers back to the custom table. For example:

<table>
<thead>
<tr>
<th>Table</th>
<th>Column name</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_organization</td>
<td>parent</td>
<td>reference</td>
<td>u_organization</td>
</tr>
</tbody>
</table>

3. Select the custom the table from the list of tables in the New Domain Table list.
Installed with domain separation

Several types of components are installed with domain separation.

Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain_admin</td>
<td>Can create, edit, and delete domains.</td>
</tr>
</tbody>
</table>

Additions to [sys_domain] fields

The sys_domain field is added to the following tables:

Table 615: Tables with the sys_domain field

<table>
<thead>
<tr>
<th>Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys_attachment</td>
</tr>
<tr>
<td>sys_user_has_role</td>
</tr>
</tbody>
</table>
### Field for the Task Table

MSP Extensions add a task_for field to the Task table to support the New Ticket module. This reference field refers to the User table.

![Dictionary Entries](image)

**Figure 751: The task_for column on the Task table**

### Options for the Group Type

MSP Extensions add several new default options to the type field of the Group table. Add to or update these types as needed to support your domains.

<table>
<thead>
<tr>
<th>Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
</tr>
<tr>
<td>Support</td>
</tr>
<tr>
<td>Visibility</td>
</tr>
</tbody>
</table>
## Business rules

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain - Activate/Deactivate</td>
<td>core_company</td>
<td>Activates the related domain if at least one of its companies is active. Deactivates the related domain if all related companies are inactive.</td>
</tr>
<tr>
<td>Domain - Cascade Company</td>
<td>core_company</td>
<td>Keeps a company’s domain in sync with its users, groups, departments, and locations.</td>
</tr>
<tr>
<td>Domain - Cascade Domain - Email</td>
<td>sys_email</td>
<td>Keeps an email’s domain in sync with its attachments.</td>
</tr>
<tr>
<td>Domain - Cascade Domain - Group</td>
<td>sys_user_group</td>
<td>Keeps a group’s domain in sync with its inherited roles (sys_group_has_role records).</td>
</tr>
<tr>
<td>Domain - Cascade Domain - Knowledge</td>
<td>kb_knowledge</td>
<td>Keeps a knowledge article’s domain in sync with its related feedback.</td>
</tr>
<tr>
<td>Domain - Cascade Domain - Task</td>
<td>task</td>
<td>Keeps the domain in sync with related tasks for wf_context, wf_executing, wf_history, attachments, emails, task_sla and its workflow, sysapproval_approver and its workflow, sysapproval_group and its workflow.</td>
</tr>
<tr>
<td>Domain - Cascade Domain - User</td>
<td>sys_user</td>
<td>Keeps a user’s domain in sync with its group membership (sys_user_grmember) and role (sys_user_has_role) records.</td>
</tr>
<tr>
<td>Domain - Cascade Domain - Version</td>
<td>wf_workflow_version</td>
<td>Keeps domains in sync with related workflow versions for wf_activity and wf_transition.</td>
</tr>
<tr>
<td>Domain - Deactivate Companies</td>
<td>domain</td>
<td>Deactivates related companies if a domain is deactivated.</td>
</tr>
<tr>
<td>Domain - Default - Task</td>
<td>task</td>
<td>Sets the task domain based on the Task for user’s domain. If this domain would be global, sets domain to Default instead.</td>
</tr>
<tr>
<td>Domain - Default - User</td>
<td>sys_user</td>
<td>Sets a user’s domain to Default if the domain otherwise would have been global.</td>
</tr>
<tr>
<td>Domain - Disallow Global Domain Record</td>
<td>domain</td>
<td>Prevents creation of a domain with the name global.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Domain - Override Copy</td>
<td>sys_app_application</td>
<td>When an application is overridden for a domain, creates a copy of its modules for the new application.</td>
</tr>
<tr>
<td>Domain - Override Copy</td>
<td>sys_data_policy2</td>
<td>When a data policy is overridden for a domain, creates a copy of its data policy rules for the new data policy.</td>
</tr>
<tr>
<td>Domain - Override Copy</td>
<td>sys_gauge</td>
<td>When a gauge is overridden for a domain, creates a copy of its gauge counts for the new gauge.</td>
</tr>
<tr>
<td>Domain - Override Copy</td>
<td>sys_ui_action</td>
<td>When a UI action is overridden for a domain, creates a copy of its UI action views for the new UI action.</td>
</tr>
<tr>
<td>Domain - Override Copy</td>
<td>sys_ui_list_control_embedded</td>
<td>When an embedded list control is overridden for a domain, creates a copy of its client and server scripts for the new embedded list control.</td>
</tr>
<tr>
<td>Domain - Override Copy</td>
<td>sys_ui_policy</td>
<td>When a UI policy is overridden for a domain, creates a copy of its UI policy actions for the new UI policy.</td>
</tr>
<tr>
<td>Domain - Set Domain - Approvals</td>
<td>sysapproval_approver</td>
<td>Sets the domain based on that of the record being approved.</td>
</tr>
<tr>
<td>Domain - Set Domain - Attachment</td>
<td>sys_attachment</td>
<td>Sets the domain based on the parent record's domain.</td>
</tr>
<tr>
<td>Domain - Set Domain - CMDB_CI</td>
<td>cmdb_ci</td>
<td>Sets a CI's domain to that of its company.</td>
</tr>
<tr>
<td>Domain - Set Domain - Department</td>
<td>cmn_department</td>
<td>Sets a department's domain to that of its company.</td>
</tr>
<tr>
<td>Domain - Set Domain - Domain</td>
<td>domain</td>
<td>Sets a domain's domain to itself.</td>
</tr>
<tr>
<td>Domain - Set Domain - Email</td>
<td>sys_email</td>
<td>Sets the domain based on the parent record's domain. An email's parent record is the record specified in the instance field.</td>
</tr>
<tr>
<td>Domain - Set Domain - Feedback</td>
<td>kb_feedback</td>
<td>Sets a knowledge feedback's domain to that of its knowledge article.</td>
</tr>
<tr>
<td>Domain - Set Domain - Group</td>
<td>sys_user_group</td>
<td>Sets a group's domain to that of its company.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Domain - Set Domain - Group Approvals</td>
<td>sysapproval_group</td>
<td>Sets the domain based on that of the record being approved.</td>
</tr>
<tr>
<td>Domain - Set Domain - Group Role</td>
<td>sys_group_has_role</td>
<td>Sets a group role's domain to that of its group.</td>
</tr>
<tr>
<td>Domain - Set Domain - Location</td>
<td>cmn_location</td>
<td>Sets a location's domain to that of its company.</td>
</tr>
<tr>
<td>Domain - Set Domain - Task SLA</td>
<td>task_sla</td>
<td>Sets a task SLA's domain to that of its task.</td>
</tr>
<tr>
<td>Domain - Set Domain - User</td>
<td>sys_user</td>
<td>Sets a user's domain to that of its company.</td>
</tr>
<tr>
<td>Domain - Set Domain - User Role</td>
<td>sys_user_has_role</td>
<td>Sets a user role's domain to that of its user.</td>
</tr>
<tr>
<td>Domain - Set Domain - WF Activity Hist</td>
<td>wf_history</td>
<td>Sets the workflow activity history domain based on the parent workflow context's domain.</td>
</tr>
<tr>
<td>Domain - Set Domain - WF Context</td>
<td>wf_context</td>
<td>Sets the workflow context domain based on the referenced record's domain, if it has one.</td>
</tr>
<tr>
<td>Domain - Set Domain - WF Exec Activity</td>
<td>wf_executing</td>
<td>Sets the workflow executing activity domain based on the parent workflow context's domain.</td>
</tr>
<tr>
<td>Domain - Set task for - Change</td>
<td>change-request</td>
<td>When converting a ticket to a change request, sets the Requested by field to the ticket's Task for value.</td>
</tr>
<tr>
<td>Domain - Set task for - Incident</td>
<td>incident</td>
<td>When converting a ticket to an incident, sets the Caller field to the ticket's Task for value.</td>
</tr>
<tr>
<td>Domain - Validate Default</td>
<td>domain</td>
<td>Ensures only one domain has the Default check box selected.</td>
</tr>
<tr>
<td>Domain - Validate Primary</td>
<td>domain</td>
<td>Ensures only one domain has the Primary check box selected.</td>
</tr>
</tbody>
</table>

**Business Rules Installed with Domain Support Plugin**

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Domain Set</td>
<td>sys_dictionary</td>
<td>Sets the domain set to the current domain.</td>
</tr>
<tr>
<td>Domain support properties</td>
<td>sys_properties</td>
<td>Sets the system properties to match the domain query method (domain paths or domain numbering).</td>
</tr>
</tbody>
</table>
Client scripts

<table>
<thead>
<tr>
<th>Client script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain - Set Company and Location</td>
<td>Monitors the incident caller field for changes. If the company and location fields do not already have a value, the script adds this information from the caller record. If the company and location fields already have a value, the script retains the existing values.</td>
</tr>
<tr>
<td>(sys_script)</td>
<td></td>
</tr>
<tr>
<td>(BP) Set Location to User</td>
<td>Monitors the incident location field and sets the location field to the caller's location.</td>
</tr>
</tbody>
</table>

Deactivated script

Domain separation administration

Administrators can view information about domain separation, identify potential issues, and change configuration settings.

Validate the domain tree

By default, the instance validates the domain hierarchy every time you change the domain table, change the query method, or reset the records to the global domain.

Role required: admin

The Domain Progress Workers list displays any currently running domain tasks. Use the following procedure to manually start the validation process.

1. Navigate to Domain Admin > Configuration.
2. From Domain Validation, click More > Validate domains hierarchy.
3. After the validation process completes, review the Domain Alerts section for any renumbering or path conversion errors.

   The domain validation process automatically fixes some validation errors and provides information about errors that cannot be automatically fixed.
View the domain log

If the domain validation indicates errors, you can see the affected domain records by navigating to the Domain Log [syslog_domain] table.

Click a domain log record to see details.
<table>
<thead>
<tr>
<th>Created</th>
<th>Level</th>
<th>Message</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-02-25 13:30:30</td>
<td>Error</td>
<td>2 records detected in 'sys_s' that are not in any existing domain and DomainPaths. Fix the domain value in these records &amp; run validator again.</td>
<td>com.glide.domain.validator</td>
</tr>
<tr>
<td>2016-02-25 13:33:19</td>
<td>Error</td>
<td>2 records detected in 'sys_s' that are not in any existing domain and DomainPaths. Fix the domain value in these records &amp; run validator again. Entries causing the error are as follows: sys_id:000000000916587707805125111111, sys_domain:sys_s</td>
<td>com.glide.domain.validator</td>
</tr>
<tr>
<td>2016-02-25 12:20:51</td>
<td>Information</td>
<td>Domain validation completed with no issues</td>
<td>com.glide.domain.validator</td>
</tr>
<tr>
<td>2016-02-25 12:21:50</td>
<td>Information</td>
<td>Domain validation completed with no issues</td>
<td>com.glide.domain.validator</td>
</tr>
<tr>
<td>2016-02-25 12:21:50</td>
<td>Information</td>
<td>Updated table sys_ui for domain: global, # of rows affected: 1158</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:22:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: Cisco, # of rows affected: 12, remaining tables: 12.</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: Cisco, # of rows affected: 3</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: global, # of rows affected: 12, remaining tables: 12.</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: global, # of rows affected: 50</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: OceanAir, # of rows affected: 12, remaining tables: 12.</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: OceanAir, # of rows affected: 3</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: inherit, # of rows affected: 12, remaining tables: 12.</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: inherit, # of rows affected: 3</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: ACME, # of rows affected: 12, remaining tables: 12.</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: ACME, # of rows affected: 3</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: global, # of rows affected: 11, remaining tables: 11.</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user for domain: global, # of rows affected: 36</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user_group for domain: global, # of rows affected: 10, remaining tables: 10.</td>
<td>com.glide.domain.paths</td>
</tr>
<tr>
<td>2016-02-25 12:23:56</td>
<td>Information</td>
<td>Updated table sys_user_group for domain: global, # of rows affected: 22</td>
<td>com.glide.domain.paths</td>
</tr>
</tbody>
</table>
View a list of tables using domain separation

You can view a list of all domain-separated tables from the Configuration module.

Role required: admin

1. Navigate to Domain Admin > Configuration.
2. From Domain Validation, click More > Show tables with Domain field.

Ignore data and proceed

When you set up domain separation, you can ignore data, which has an affect on domain visibility settings, overrides, and contains definitions.

Several changes occur when you select Ignore All Data and Proceed

- The domain table changes to the table you selected.
- All domain visibility settings (both user and group) are deleted.
- All existing records with a domain value refer to invalid domains until you migrate the domain data.
- All existing domain overrides refer to invalid domains until you migrate the domain data.
- All existing domain contains definitions refer to invalid domains until you migrate the domain data.

**Note:** Visibility settings are deleted whenever the domain table reference changes.

When you select the ignore option, no existing domain-separated tables are moved to the global domain, and it is your responsibility to migrate the domain records. Until the migration is complete, the domain validator will show warnings about inconsistent domain data. If necessary, you can manually reset all domain-separated tables to the global domain.

Reset all records to the global domain

You can manually reset all domain-separated records to the global domain at any time.

Role required: admin

**Warning:** The only way to recover resetting records to the global domain is to restore from a data back-up.

1. Navigate to Domain Admin > Configuration.
2. From Domain Validation, click More > Reset all records to Global.

Manually re-enable delegated administration

Delegated administration allows administrators in lower portions of the domain hierarchy to add domain-specific policies that override policies set higher in the domain hierarchy.

Role required: admin

By default, activating domain separation enables delegated administration. Use the following steps to manually re-enable delegated administration if it was previously disabled.

1. Navigate to Domain Admin > Configuration.
2. For Enable delegated administration, select the Yes check box.
3. Click Save.

Manually re-enable domain separation

Use the following steps to manually re-enable domain separation if it was previously disabled.
Role required: admin

1. Navigate to **Domain Admin > Configuration**.
2. Select the domain table. For example, to navigate to the Group \[sys_user_group\] table, click **User Administration > Groups**.
3. Select the domain query method. For example, **Switch to Domain Paths**.
4. For **Enable domain separation**, select the **Yes** check box.
5. Click **Save**.

**Disable and re-enable domain numbering**

Your instance maintains domain numbers even after you install domain paths so that you have to the option to switch back and forth between paths and numbers during a validation period.

Role required: admin

Since maintaining domain numbers requires additional overhead, disable numbers after you successfully test and validate the domain paths query method.

---

**Note:** Domain numbers are no longer supported.

To disable domain numbering updates, add the system property `glide.sys.domain.numbers.installed` and set it to `false`. This property prevents the instance from making any further updates to the `sys_domain_number` and `sys_domain_increment` fields.

You can re-enable the domain numbering if you must revert query methods for any reason.

1. Navigate to **Domain Admin > Configuration**.
2. From **Query Method**, click **More > Show tables with Domain field**.
3. Click **OK** to confirm the selection.

**Troubleshoot domain separation problems**

You can use the following information to troubleshoot problems with domain separation.

Role required: admin

1. Navigate to **Domain Admin > Configuration**.
2. For **Enable verbose domain logging**, select the **Yes** check box.
3. Click **Save**.

**Enable domain logging and debug messages**

Domain log and debug messages allow you to troubleshoot domain configuration errors.

Role required: admin

1. Navigate to **Domain Admin > Configuration**.
2. For **Enable verbose domain logging**, select the **Yes** check box.
3. Click **Save**.

View a real-time domain message

You can view real-time domain messages from the system logs.

Role required: admin

1. Enable verbose domain logging.
2. Navigate to **System Diagnostics > Session Debug > Enable All**. Because this is a real time review, there is no need to let the debug session run for a time before checking the log files.
3. Navigate to **System Logs > System Log > All.**
4. Search for the text Query against table.

This query finds log messages in this format:

```
08:36:43.974: [Domain Spool] Query against table incident restricted by domain values [Database Atlanta[db53580b0a0a0a6501aa37c294a2ba6b], Database[287ee6f6a919fe198100ada7950d0b1b73], Database San Diego[db53a9290a0a650091abebeccf833c6], global, NY DB[5f74727dc0a8010e01efe33a29193f9]]
```

In this example, the user viewing the Incident table only saw records that matched the Database Atlanta, Database, Database San Diego, global, and NY DB domains.

View a historical domain message

You can view historical domain messages in the log file to troubleshoot domain separation issues.

Role required: admin

1. Enable verbose domain logging.
2. Navigate to **System Diagnostics > Session Debug > Enable All.**
3. Let the debug session run for a time period, such as a day, before checking the log files.
4. Navigate to **System Logs > Utilities > Node Log File Download.**
5. Open the record for the day you want to view. Log files use the naming format `localhost_log.<yyy-mm-dd>.txt`.
6. Click the Download log related link.
7. Open the downloaded log file in a text editor and search for log messages with the following format:

```
Query against table incident restricted by domain values [global, Software[8a4dde73c6112278017a6a4baf547aa7]]
```

In this example, a user only saw records from the Incident table that matched the global and Software domains.

View domain relationships

The domain map offers domain administrators a read-only representation of the active domains on the instance and how they relate to each other.

Role required: admin

All domain maps must have one domain set as the primary domain. In addition, each domain in the domain map must meet these criteria:

- The **Parent** field must be filled in (the primary domain is the only exception to this).
- The **Active** check box must be selected.

The domain map does not draw domain relationships for domains that fail to meet the mapping criteria.

1. Navigate to **Domain Admin > Domain Map.**
2. Click the plus (+) or minus (-) icons on the domain headers to show or hide sub domains.

**Troubleshoot domain separation errors**

If you encounter domain separation issues, review this list of solutions.
<table>
<thead>
<tr>
<th>Error or symptom</th>
<th>Solution</th>
</tr>
</thead>
</table>
| A domain sys_id points to a non-existent domain      | This error occurs when a data record, such as a user or task record, has a sys_domain column value whose sys_id does not exist in the current domain table. The domain sys_id could have been accidentally deleted or it could refer to a previous domain table if you changed the domain table. To fix the error, open a list for the table containing the error, filter on the invalid sys_domain value. Then, either manually enter the correct sys_domain value or remove it.  
  **Note:** You can have invalid domain sys_ids in any table that references the domain table. For example, invalid domain IDs can occur in the User Visibility Domain [sys_user_visibility], Group Visibility Domain [sys_user_group_visibility], and Contained Domain [domain_contains] tables. |
| A domain path or domain number sys_id points to the wrong domain | This error occurs when a domain number or domain path query is out of sync with the actual domain name. This error can occur with domain numbers when adding domains requires renumbering or during the conversion from domain numbers to domain paths. To fix the error, validate the domain tree and let the background conversion process run to completion. If the error persists, you can manually edit the value for the sys_domain_path or sys_domain_number columns to point to the proper domain. |
| The domain tree structure is corrupt                 | This error occurs if there is a series of domain contains relationships that create an infinite loop among domains. To fix the error, open a list for the domain table and manually edit the domain contains values to not form a loop. |

**ServiceNow Edge Encryption**

With Edge Encryption, you can control and possess all encryption keys for encrypted data.

Edge Encryption is a proxy application that resides in your network and encrypts data before the data is sent over the Internet to your instance (encrypted in motion). The data remains encrypted while stored in the instance (encrypted at rest). The encrypted data is sent back to the proxy application (encrypted in...
motion) when requested. Finally, the encrypted data is decrypted by the proxy before being sent to the client in your network.

Your security administrator specifies which fields are to be encrypted. AES 128 or AES 256 encryption algorithms can be used. Attachments can be encrypted on a table by table basis.

Depending on the encryption type chosen for a field, certain levels of UI filtering, sorting, or compare functionality can be preserved.

![Edge Encryption Diagram]

**Figure 753: Edge Encryption**

You own and manage the encryption keys. Encryption keys are never sent to the instance. ServiceNow never possesses the clear data and cannot see it. Three key storage mechanisms are supported: file store, Java KeyStore, and SafeNet. The Edge Encryption proxy obtains encryption keys from one of the key stores to encrypt and decrypt data.

**Edge Encryption limitations**

Edge Encryption impacts system functions. Carefully evaluate the impact of encrypting a field.

**Field type restrictions**

Restrictions on encrypting field types.

- Only string fields can be encrypted. Choice fields, virtual fields, journal fields, and any fields other than string fields cannot be encrypted. See **Field types** on page 777 for more information.
- Fields in system tables, except for certain fields in sys_user, cannot be encrypted.
- System fields in tables cannot be encrypted.
- Fields named "number" and fields associated with an auto-numbering scheme cannot be encrypted.
- Encrypted fields are not available in **Go to** and header filter boxes.
- When encrypting fields used as an index, only order preserving and equality preserving encryption types can be used. Indexed fields cannot be encrypted using the standard encryption type.
Configuration restrictions

Restrictions and behavior of encryption configurations.

• After a field has been added to the Edge Encryption Configuration table, the configuration record cannot be deleted. If you no longer want a field to be encrypted, deactivate the record in the Edge Encryption Configuration table, and schedule an encryption job to decrypt the data.

• If a field in a parent table is marked to be encrypted, the field in all inherited tables is also encrypted. For example, if the short description field in the Task table is encrypted, then the contents of the short description field in the Incident table are encrypted.

• If a field inherited from a parent table is marked to be encrypted, the field in the parent table cannot be encrypted. For example, if short description in the Incident table is marked to be encrypted, then short description in the Task table cannot be encrypted. In the example, you can encrypt the short description in the Problem table.

• When a field with an encryption configuration defined is exported to any format, the output includes encrypted values even when exported through the proxy server. Importing data to a field with an encryption configuration defined is not supported.

Instance restrictions

Impact of using Edge Encryption on the instance.

• Back-end logic cannot process encrypted data. When the instance contains encrypted data, any business rule, back-end script, or back-end feature that relies on evaluating the data in the encrypted field does not run correctly.

• Scripts run on the server cannot change encrypted data.

• Global search is not supported. Because global search attempts to search both encrypted and clear text data, the results may not be what the user expects.

• Encrypted data cannot be copied to a record where the field is not encrypted.

• Depending on the type of encryption selected, the user interface functionality for the encrypted fields is reduced. For example, being able to compare, group by, sort, and search may be impacted. Generally, the stronger the encryption selected, the more functionality is reduced.

• Other than file store, Java KeyStore, and SafeNet, no third-party software, or hardware encryption key management is supported.

• While multiple encryption proxies connected to a single instance are supported, encryption proxy cluster management and monitoring are not available. Each proxy must be managed separately.

• System configurations such as workload and the number of encrypted fields can impact the performance of encrypted fields.

• The Edge Encryption proxy server can only connect to a single instance

• If your instance uses an Oracle database and the string field you are marking to be encrypted is greater than 2925 characters, that field cannot be sorted even when order preserving encryption is selected.

• If your instance uses an Oracle database, Unicode AL32UTF8 is the only supported character set.

• Encrypted values included in emails are encrypted.

Getting started with Edge Encryption

Successful implementation of Edge Encryption requires planning and preparation.

Answer the following questions in the planning stage.

• Which fields are to be encrypted?

• Which encryption types are to be used?

• How many Edge Encryption proxies are needed?
• If an order preserving encryption type is to be used, where will the MySQL instance be located?
• Which key management system is to be used?
• How are browsers to be set up to access an Edge Encryption proxy?

System administrators, network administrators, and security team members have different tasks to fulfill for implementing Edge Encryption.

• System administrators need the security-admin role. The system administrator needs to:
  • Download the Edge Encryption proxy application.
  • Set up an Edge Encryption user account for the proxies to use to connect to the instance. The user must be assigned the edge_encryption role.
  • Configure Edge Encryption on the instance.
  • Schedule encryption jobs.
  • Monitor Edge Encryption.
  • Create and edit encryption rules.

• Your network administrator needs to:
  • Install the Edge Encryption proxy application.
  • Know network addresses for proxy servers, and the database server used with order-preserving encryption.
  • Install the database to be used with order-preserving encryption.
  • Start and stop the proxy applications.
  • Perform encryption key management.
  • Determine how to map users to encryption proxy applications. This can be done with DNS settings or routing rules, and is specific to each network.
  • Deal with the issues of using multiple encryption proxy applications.

• Your security administrator must determine the encryption types to be assigned to each field.

Edge Encryption application and proxy

Edge Encryption has these components: the Edge Encryption applications installed via a plugin and the Edge Encryption proxy which can be downloaded from one of the menu options from the Edge Encryption application.

Use the Edge Encryption application to specify which fields and attachments are encrypted, manage encryption rules, and schedule mass encryption jobs.

The Edge Encryption proxy uses encryption rules to identify in an HTTP request what, if anything, needs to be encrypted and encrypts it before forwarding the request to the instance. For decryption, the Edge Encryption proxy looks at the HTTP responses for any encrypted data and decrypts it before sending the response back to the client. In order for this to happen, all HTTP requests and responses must go through the Edge Encryption proxy. This includes any requests originating from a browser, as well as any SOAP and REST requests.

Encryption types

Edge Encryption provides a variety of encryption types. For each encryption type, it provides support for AES with 128-bit encryption keys. If the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy files are installed, it also provides support for 256-bit encryption keys for each of the encryption types.

The following encryption types are listed in decreasing security quality.
Table 616: Encryption types

<table>
<thead>
<tr>
<th>Encryption type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard AES 256</td>
<td>Fields cannot be filtered, sorted, or compared.</td>
</tr>
<tr>
<td>Standard AES 128</td>
<td>Fields cannot be filtered, sorted, or compared.</td>
</tr>
<tr>
<td>Equality preserving AES 256</td>
<td>Fields can be filtered using equality comparisons.</td>
</tr>
<tr>
<td>Equality preserving AES 128</td>
<td>Fields can be filtered using equality comparisons.</td>
</tr>
<tr>
<td>Order preserving AES 256</td>
<td>Fields can be sorted and equality comparison filtering can be used. Requires the use of a MySQL database in your network.</td>
</tr>
<tr>
<td>Order preserving AES 128</td>
<td>Fields can be sorted and equality comparison filtering can be used. Requires the use of a MySQL database in your network.</td>
</tr>
</tbody>
</table>

When using standard encryption, the encrypted value of a field is different each time the field is encrypted, even when the field value remains the same. Standard encryption is the most robust form of encryption. Fields using standard encryption cannot be sorted, grouped by, or filtered on.

When using equality preserving encryption, the encrypted value of a field is the same when the field value remains the same. When a field is encrypted using equality preserving encryption, it is possible to perform equality comparisons and group by operations on that field.

**Note:** When equality preserving encryption is selected for a field that already contains data, performing a group by action on the field may not group together fields with the same value if one is encrypted and the other is not.

When a field is encrypted using order preserving encryption, in addition to supporting all the features supported by equality preserving, it also supports the ability to sort data in the field. The order preserving encryption type is only supported if there is a MySQL database configured for the Edge Encryption Proxy.

**Note:** When using order preserving encryption and the proxy database is down, updates can be made to fields using order preserving encryption. However, the sort order will not be correct when trying to sort data based on those fields. Group by will also not work as expected. When the database is again operational, schedule an order token repair job to repair missing tokens.

Encrypted attachments

You can encrypt attachments for specific tables.

All attachments to a table use the same encryption type. Encrypted attachments are not searched when performing a text search. Only the standard encryption types are allowed for attachments. The order preserving or equality preserving encryption types are not allowed.

For a session bypassing the Edge Encryption proxy:

- On a record with attachment encryption enabled:
  - The user can see that there are attachments and the attachment names.
  - The user cannot open or download the attachments.
  - The user cannot add new attachments.
On a record without attachment encryption enabled:
- The user can open and download existing attachments.
- The user can add new attachments.

For a session using the encryption proxy, the user can open and download existing attachments and add new attachments.

Key management

You are responsible for providing and managing the encryption keys used by Edge Encryption.

When obtaining and creating encryption keys to support the encryption types used by Edge Encryption, you should consider the following:
- Whether to use AES 128 or AES 256. You must define a default AES 128 encryption key even if it is not used.
- Whether to use file store, Java KeyStore, or NAE.
- When to rotate encryption keys.
- When and if to use a mass encryption job to re-encrypt data using the new key.

Each key is defined by a set of properties. These properties must be the same in every proxy configuration file. When adding a new key, also add the set of properties to each proxy configuration file. Removing the set of properties also removes the key from use. Before removing a key from the proxy configuration files and the key store, decrypt all data on the instance that uses the key. You can do this by adding a new encryption key and scheduling a single key rotation job.

A number is appended to the property to make the property unique. For example, edgeencryption.encrypter.type.1. The group of properties for a key must have the same number appended.

Key store management

Encryption keys must be stored in one or more encryption key stores.

Edge Encryption supports the following types of key storage mechanisms.

- **File system**: Keys are stored in a file in a file system that is accessible by the Edge Encryption proxy. Encryption keys stored in a file are not encrypted so it is your responsibility to protect these files.
- **Java KeyStore**: Keys are stored in Java's JCEKS keystore. A Java Keystore is protected by a password so it is more secure than storing keys in a file in the file system. A single Java Keystore can store multiple keys and the keys are identified by a key alias, making it easier to manage multiple keys.
- **NAE (Network Attached Encryption) key store**: Keys are stored and retrieved with SafeNet's KeySecure key management.

The Edge Encryption proxy ships with the Java JCEKS Keystore file named keystore.jceks in the keystore directory. This keystore file contains the ServiceNow certificate used to validate encryption rules that have been signed by ServiceNow.

**Note**: If using a keystore other than the base system JCEKS KeyStore, you must import the ServiceNow public key into your keystore. The public key alias is servicenow.

In addition to the encryption keys, the Java JCEKS Keystore is used to store the RSA key pair for digitally signing the encryption configuration and encryption rules that are stored in the instance, and the digital certificate that the Edge Encryption proxy uses to establish a secure connection with the browsers and any other clients.
Request Edge Encryption

The Edge Encryption plugin is available as a separate subscription.

Role required: none

To purchase a subscription, contact your ServiceNow account manager. The account manager will arrange to have the plugin activated on your organization’s production and sub-production instances, generally within a few days.

If you do not have an account manager, decide to delay activation after purchase, or want to evaluate the product on a sub-production instance without charge, follow these steps.

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
<tr>
<td>Specify the date and time you would like this plugin to be enabled</td>
<td>Date and time must be at least 2 business days from the current time.</td>
</tr>
<tr>
<td>Reason/Comments</td>
<td>Any information that would be helpful for the ServiceNow personnel activating the plugin such as if you need the plugin activated at a specific time instead of during one of the default activation windows.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

Edge Encryption proxy installation

Install one or more Edge Encryption proxy applications on your network.

Installing a Edge Encryption proxy includes these steps.

- Install the Edge Encryption proxy application on a server in your network.
- Generate the RSA key pair for digitally signing encryption configurations and encryption rules.
- Install the Java Cryptography Extension (JCE), if you plan to use AES 256 encryption.
- If you are using a secure SSL connection, obtain a server certificate and import it to the Java KeyStore key store.
- If order preserving encryption types are to be used, set up a MySQL database instance on a machine in your network.
- Set up the `edgeencryption.properties` configuration file.
- Set up each user’s browser to point to an Edge Encryption proxy.
Accessing the proxy server

Once installation is complete, point each user’s browser to an Edge Encryption proxy using the URL format: `<host>:<port>`. Values are determined by the host and port properties in the `edgeencryption.properties` file. See Configure the proxy properties on page 2651.

As an example with the following values:

<table>
<thead>
<tr>
<th>Property</th>
<th>Example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.proxy.host</td>
<td>hostname.mycompany.com</td>
</tr>
<tr>
<td>edgeencryption.proxy.http.port</td>
<td>8081</td>
</tr>
</tbody>
</table>

A client will access the proxy server using the following address: `http://hostname.mycompany.com:8081/`.

**Note:** DNS settings and routing rules may be used. Host and port values are determined by your network administrator.

Edge Encryption system requirements

The Edge Encryption proxy application can run on servers or virtual machines running Windows or Linux.

The Edge Encryption proxy distribution comes with JRE 1.8. Out of the box, the Edge Encryption proxy uses the JRE bundled with the distribution. You can change this by modifying the `wrapper.conf` file in the `<installation dir>/conf` directory.

Proxy server minimum configuration

The minimum configuration includes:

- A minimum of 4 GB of available RAM per proxy server (6 GB is recommended for most deployments).

  **Note:** The host machine running the proxy server must have at least 1 more GB of RAM available for OS services than is needed to run the proxy server. Specifically, if a proxy server is set to use 4 GB of RAM, the VM or hardware must have at least 5 GB of RAM installed.

- 3+ GHz CPU, with a 4-core CPU preferred.
- Multiple proxy servers behind a load balancer. The number of proxy servers will depend on the number of application nodes, number of simultaneous users, and failover needs. See Sizing your Edge Encryption environment on page 2646 for considerations.
- Ability to ride-along with other services, depending on the server utilization and resource availability.

Proxy server supported systems

The following systems are supported.

- Windows Server
  - All Windows Server 2012-R2 editions or later
  - Virtual machines or physical hardware
  - 32-bit and 64-bit systems
- Linux
• Virtual machines or physical hardware
• 32-bit and 64-bit systems

**Note:** Because the proxy server requires access to at least 4 GB of RAM, a 64-bit operating system and JVM is recommended.

On 64-bit Linux systems, you must install the 32-bit *GNU C library* (glibc). The installation command for CentOS is: `yum install glibc.i686`

**Proxy database minimum configuration**

Order preserving encryption requires a MySQL database configured for the Edge Encryption proxy server. Use a dedicated machine to run the proxy database. The minimum requirements include:

- **Version:** MySQL database versions 5.5 and higher
- **OS:** 64-bit systems
- **CPU:** 2+ GHz CPU, with a 4-core CPU preferred
- **RAM:** 16 GB
- **Disk:** SAN or local storage (RAID 10 recommended)
- **Size:** Determined by the number of potential records multiplied by record size. *Calculate the proxy database size* on page 2647.
- **Configuration:** High Availability cluster. If you are unsure of how to configure your MySQL server, contact MySQL for configuration information.

*Sizing your Edge Encryption environment*

Choosing the number of proxy servers for your environment is an important task. Consider the number of users, redundancy needs, and acceptable latency.

**Redundancy**

Maintain redundant proxy servers in case of hardware failure. Proxy servers should be located behind a load balancer to provide a functional path for all users if a proxy server is unreachable. At a minimum, ensure that two proxy servers are always available.

**Size**

Size refers to the number of proxy servers required to avoid additional latency that the encryption of data produces. Depending on use, you may want to reduce the amount of latency by adding additional proxy servers. For example, if regular mass encryptions are run, add additional proxy servers to handle the load, or run the mass encryptions when the user load is light. In addition, the hardware that the proxy server runs on influences performance. Proxy servers running on hardware with faster CPUs, more CPUs, and more RAM have higher throughput than slower, limited systems.

The following guidelines assume that your proxy server is running on at least the minimum hardware requirements. To determine the number of proxy servers:

- Consider setting up one proxy server for every two application nodes on the instance.
- For redundancy, set up a minimum of two proxy servers behind a load balancer.
- Add an extra proxy server for every 500 simultaneous users.
- Depending on the desired redundancy, add additional proxy servers for failover.
For example, for an instance with 2,000 users, you should have at least five proxy servers behind a load balancer. This calculation includes one proxy server for every 500 users, with an extra proxy server for failover. Determine ahead of time when you will approach a threshold of 500 users and place another proxy server in the load balancer pool.

**Load balancers**

To balance requests and improve server response time, distribute proxy servers in a load balancer pool. Configure load balancers to use the "least connections" method. This method connects requests to the proxy server with the fewest active connections, preventing the overloading of a single proxy.

**CPU utilization**

Because data encryption and tokenization are CPU intensive operations, CPU spikes while encrypting data are normal and expected. When CPU utilization is over 80% for several minutes at a time, it likely means that the proxy server has too much work to do. When this happens, latency increases for the period that the CPU utilization is high. If latency persists, adding another proxy server may help decrease the latency.

**Memory**

The proxy server must have a minimum of 4 GB of RAM available (6 GB recommended). Set the proxy server initial and upper bound memory limits to the recommended settings.

*Calculate the proxy database size*

If using order-preserving encryption, determine the size of your MySQL database by multiplying the number of potential records by record size.

**Note:** Use a dedicated machine to run the proxy database. Do not run the database on the same hardware as the proxy server.

1. Determine the potential number of records that could include fields encrypted with order-preserving encryption.
   a) Multiply the number of encryption configurations using order-preserving encryption by the number of records each configuration is applied to.
   b) To allow for growth, multiply the result by three.

2. Multiply the result of step 1 by 1,536.
   1,536 is the average size of a record in bytes.

The calculated value is the recommended size in bytes for your proxy database.

**Encryption proxy connection requirements**

The proxy server that will run the Edge Encryption application must be able to communicate with machines in your network.

Make sure that the proxy server has these network privileges:

- **Firewall access:** Configure any firewalls between the proxy server and the client devices to allow a connection. If your network uses a DMZ, and if your network security protocols limit port access from within the network to the DMZ, you might have to deploy a proxy server to a machine within the DMZ.
- **Network access**: Configure each client to let the proxy server connect with it. If network security prevents you from configuring new machines that can connect to the clients, install the proxy server on an existing machine with connection privileges.

- **Network account**: Install the proxy server with the proper account, either local or domain administrator.

Additionally, for the proxy server to access your instance:

- Ensure network access to the instance. Make sure that the network used by the proxy server is configured to allow traffic over TCP port 443.

---

**Download the Edge Encryption proxy application**

Download the Edge Encryption proxy application from your instance, and then copy the ZIP file to each computer that is to run the Edge Encryption proxy server.

Before starting this procedure, the Edge Encryption plugin must be installed and activated on your instance.

Role required: security-admin

1. Navigate to **Encryption Configuration > Installation & Downloads > Downloads**.
2. Select the OS version for your proxy server.
3. Copy the ZIP file to each computer that is to run the Edge Encryption proxy server.

---

**Install the encryption proxy on Linux**

You can install an Edge Encryption proxy on a 32-bit or 64-bit Linux computer.

1. Create the installation directory.
2. Copy the Edge Encryption archive file to the installation directory.
3. Change to the installation directory.
4. Run the following command:

   ```bash
   java -jar edgeencryption-dist-<version>-linux-x86-64.zip -m install -n <ProxyName> -h <host> -p <port> -proto https
   ```

**Table 617:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version number of the Edge Encryption release.</td>
</tr>
<tr>
<td>ProxyName</td>
<td>The name of the encryption proxy instance.</td>
</tr>
<tr>
<td>host</td>
<td>Your instance.</td>
</tr>
<tr>
<td>port</td>
<td>The port of your instance. When the protocol is https, the port is normally 443.</td>
</tr>
</tbody>
</table>

If you want to see the help screen, execute this command without arguments:

```bash
java -jar edgeencryption-dist-<version>-linux-x86-64.zip
```

The **ProxyName_port** directory is created in the current directory. The **edgeencryption.properties** file is updated with the host, port, and protocol values from the command line.
Install the encryption proxy on Windows

You can install an Edge Encryption proxy on a 32-bit or 64-bit Windows computer.

1. Create the installation directory.
2. Download the Edge Encryption proxy archive file to the installation directory.
3. Start the Windows cmd terminal program with administrator privileges.
4. Change to the installation directory.
5. Run the following command:

```
java -jar edgeencryption-dist-<version>-windows-x86-64.zip -m install -n <ProxyName> -h <host> -p <port> -proto https
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version number of the Edge Encryption release.</td>
</tr>
<tr>
<td>ProxyName</td>
<td>The name of the encryption proxy instance.</td>
</tr>
<tr>
<td>host</td>
<td>Your instance.</td>
</tr>
<tr>
<td>port</td>
<td>The port of your instance. When the protocol is https, the port is normally 443.</td>
</tr>
</tbody>
</table>

If you want to see the help screen, execute this command without arguments: `java -jar edgeencryption-dist-<version>-windows-x86-64.zip`

6. Optionally change the name of the service.

**Caution:** You must perform this step before running the install command on Windows. See Start the Edge Encryption proxy on page 2667.

If this step is not performed, the Edge Encryption proxy service installs under the name Wrapper.

a) Open the `conf/wrapper.conf` file on the new proxy and add the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wrapper.ntservice.name</td>
<td>Unique name of the Edge Encryption proxy service.</td>
</tr>
<tr>
<td>wrapper.ntservice.displayname</td>
<td>Edge Encryption proxy service display name.</td>
</tr>
<tr>
<td>wrapper.ntservice.description (Optional)</td>
<td>Proxy server description.</td>
</tr>
</tbody>
</table>

b) Save and close the file.

The `ProxyName_port` directory is created in the current directory. The `edgeencryption.properties` file is updated with the host, port, and protocol values from the command line.

Configure the target properties

In the `edgeencryption.properties` file, specify which instance the Edge Encryption proxy will communicate with. These values are initially set when the proxy application is installed.

1. Change to the `<installation directory>/conf/` directory.
2. Open the `edgeencryption.properties` file.
3. Enter the following target properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.target.host</td>
<td>The host name for the instance. It must be the same for all encryption proxies. This property is set when the proxy is installed. For example, <code>instancename.service-now.com</code></td>
</tr>
<tr>
<td>edgeencryption.target.port</td>
<td>The instance port. It must be the same for all encryption proxies. This property is set when the proxy is installed.</td>
</tr>
<tr>
<td>edgeencryption.target.protocol</td>
<td>The instance protocol. It must be the same for all encryption proxies. This property is set when the proxy is installed.</td>
</tr>
</tbody>
</table>

4. Save and close the `edgeencryption.properties` file.

Set up an Edge Encryption user account

The Edge Encryption proxies connect to the instance as a user, in order to obtain and update encryption configuration information. Create a user account for this purpose and give the `edge_encryption` role to the user.

The Edge Encryption plugin must be installed before you can assign the role.

Role required: admin

1. On your ServiceNow instance, create a user account to be used by the Edge Encryption proxy applications.
2. Assign the `edge_encryption` role to the user.
3. On each proxy, update the `edgeencryption.properties` file with the properties for the user account.
   a) Change to the `<installation directory>/conf/` directory.
   b) Open the `edgeencryption.properties` file.
   c) Enter the properties for the user account.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.target.username</td>
<td>The user name that the proxy uses to log in to the instance. The user must have the <code>edge_encryption</code> role.</td>
</tr>
<tr>
<td>edgeencryption.target.password</td>
<td>The password that the proxy uses to log in to the instance.</td>
</tr>
</tbody>
</table>

4. Save and close the `edgeencryption.properties` file.
Configure the proxy properties

In the `edgeencryption.properties` file, you can specify how the Edge Encryption proxy will communicate with your ServiceNow instance.

**Note:** If you are copying an `edgeencryption.properties` file from one proxy to another, verify that the name property is unique.

1. Change to the `<installation directory>/conf/` directory.
2. Open the `edgeencryption.properties` file.
3. Enter the proxy properties.

**Table 620: Proxy properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.proxy.host</code></td>
<td>The server name, IP address, or fully-qualified domain name of the computer running the proxy. Along with the port, this property defines the URL used by the client to access the proxy server.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.name</code></td>
<td>Name of the proxy. It must be unique for each proxy.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.http.port</code></td>
<td>Port on the proxy for HTTP communication.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.https.port</code></td>
<td>Port on the proxy for HTTPS communication.</td>
</tr>
</tbody>
</table>

4. Save and close the `edgeencryption.properties` file.

Configure web proxy properties

If your network uses a web proxy, you can set up the Edge Encryption proxy to use the web proxy. This feature is available in Geneva Patch 5 and subsequent releases.

If your network does not use a web proxy, leave these configuration parameters commented out.

The Edge Encryption proxy supports HTTP connection to and basic authentication with the web proxy.

1. Change to the `<installation directory>/conf/` directory.
2. Open the `edgeencryption.properties` file.
3. Enter the proxy properties.

**Table 621: Proxy properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.webproxy.host</code></td>
<td>Can be the web-proxy's name or its IP address.</td>
</tr>
<tr>
<td><code>edgeencryption.webproxy.port</code></td>
<td>Port on the web proxy.</td>
</tr>
<tr>
<td><code>edgeencryption.webproxy.user</code></td>
<td>User name to be used to connect to the web proxy. If your web proxy does not use authentication, leave this property commented out.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>edgeencryption.webproxy.password</td>
<td>Password to use to connect to the web proxy. If your web proxy does not use authentication, leave this property commented out.</td>
</tr>
</tbody>
</table>

4. Save and close the `edgeencryption.properties` file.

5. If the web proxy requires a customer-specific client certificate, add the certificate to the Edge proxy Java truststore at `<installation directory>/java/jre/lib/security/cacerts`.
   
   To import a certificate to the truststore, execute the command:
   
   ```bash
   keytool -keystore cacerts -importcert -alias <chooseAlias> -file <certificateFile>
   ```

Install the Java Cryptography Extension (JCE)

If you want to use AES 256 encryption, you must install the Java Cryptography Extension (JCE) jurisdiction policy files and copy them into each Edge Encryption proxy.

Edge Encryption ships with the AES 128 policy files, which you must overwrite with the AES 256 policy files. It is only necessary to download the JCE once, but every Edge Encryption proxy must be updated.

1. Download the JCE policy 8 ZIP file from Oracle.
2. Unzip the file.
3. On each proxy server, copy the `local_policy.jar` and `US_export_policy.jar` files into the `<proxy-dir>/java/jre/lib/security` folder.

Create and configure the RSA key pair for the digital signature

Create an RSA key pair that the proxy can use to create the digital signature for signing changes to the encryption properties and configuration.

To generate and validate the digital signature, an RSA key pair must generated and stored in a JCEKS Java KeyStore and each proxy must be configured to use this key pair. You generate an encryption key pair by using the `keytool` command.

If the proxy was installed on SElinux (e.g. CentOS), to use the `keytool` utility you must enable loading of shared libraries from the proxy java-installation directory. To do this, run the following command as root.

```bash
chcon -R -t texrel_shlib_t proxy_install_dir/java/jre /lib
```

You must use the Java 1.8 version of the `keytool` utility. A copy of the utility can be found in `<proxy install dir>/java/jre/bin/keytool`.

1. Change to the KeyStore directory in the proxy download directory.
2. Change the default password.
   
   The default password is `changeme`.
   
   ```bash
   keytool -keystore keystore.jceks -storetype jceks -storepasswd -new <new password>
   ```
3. Create an encryption key pair.

   **Note:** Do not enter a password for the key when the `keytool` utility prompts for one.
Enter this command on a single line.

```
keytool -genkeypair -alias <key alias> -keyalg rsa -keystore keystore.jceks -storetype jceks -storepass <keystore password> -keysize 2048
```

4. Update the encryption proxy property file (`edgeencryption.properties`).
   a) Change to the `<installation directory>/conf/` directory.
   b) Open the `edgeencryption.properties` file.
   c) Enter the properties for the digital signature.
      These properties must be the same for all proxies.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.proxy.signature.keystore.path</code></td>
<td>Path and Java KeyStore file name.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.signature.keystore.password</code></td>
<td>Password. The default password is <code>changeme</code>. Change the password after installing the Java KeyStore.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.signature.keystore.keyalias</code></td>
<td>Key alias given in as the <code>-alias</code> argument when the RSA key pair was generated.</td>
</tr>
</tbody>
</table>

5. Save and close the `edgeencryption.properties` file.

Import and configure the certificate for secure SSL connection

To use a secure SSL connection, import a server certificate and add it to the Java KeyStore.

You must obtain the server certificate before you can add it to the Java KeyStore.

1. Add a server certificate to the Java KeyStore.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you have the RSA private key in the Java Keystore and generated the certificate from that key.</td>
<td><code>keytool -import -alias keyname -file server.cert -storetype JCEKS -keystore keystore.jceks -storepass pwd</code></td>
</tr>
<tr>
<td>If you have a PKCS12 file that contains the RSA key and the certificate.</td>
<td><code>keytool -importkeystore -destkeystore keystore.jceks -deststoretype jceks -srcstoretype pkcs12</code></td>
</tr>
</tbody>
</table>

Make sure the private key password is the same as the Java Keystore password.

You can run this command to change the password.

```
keytool -keypasswd -keystore keystore.jceks -alias <key alias>
```
For testing, you can use this command to generate a self-signed certificate.

```
keytool -genkeypair -alias cert -keystore keystore.jceks -storetype jceks -keyalg rsa
```

2. Update the `edgeencryption.properties` file.
   a) Change to the `<installation directory>/conf/` directory.
   b) Open the `edgeencryption.properties` file.
   c) Enter the properties for the SSL certificate.

   The certificate must be the same for all proxies.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.proxy.https.cert.alias</code></td>
<td>Alias of the certificate.</td>
</tr>
</tbody>
</table>

3. Save and close the `edgeencryption.properties` file.

Configure encryption keys using SafeNet KeySecure

If you are using a SafeNet key store, copy a set of libraries into the proxy distribution directory.

You must install and set up the SafeNet keystore before performing this step. Secure a license with Gemalto in order to download the libraries.

**Note:** On Linux, the file paths use a forward slash.

1. Copy these files to `<installation directory>/lib` directory:
   - `ingrian-nae-8.2.0.000.jar`
   - `ingrian-log4j-api-2.0-rc1.jar`
   - `ingrian-log4j-core-2.0-rc1.jar`

2. Change to the `<installation directory>/conf/` directory, and open the `edgeencryption.properties` file.

3. Enter the properties for the SafeNet key store.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.nae.retries</code></td>
<td>Number of retries to make.</td>
</tr>
<tr>
<td><code>edgeencryption.nae.enabled</code></td>
<td>If an NAE device is available.</td>
</tr>
<tr>
<td><code>edgeencryption.nae.server</code></td>
<td>Name of the NAE server.</td>
</tr>
<tr>
<td><code>edgeencryption.nae.port</code></td>
<td>Port used by the NAE server.</td>
</tr>
<tr>
<td><code>edgeencryption.nae.protocol</code></td>
<td>Protocol used by the NAE server.</td>
</tr>
<tr>
<td><code>edgeencryption.nae.keystore.path</code></td>
<td>Path to the key store on the NAE server.</td>
</tr>
<tr>
<td><code>edgeencryption.nae.keystore.password</code></td>
<td>NAE key store password.</td>
</tr>
</tbody>
</table>
An example for a SafeNet key store.

```plaintext
edgeencryption.nae.retries = 3
edgeencryption.nae.enabled = true
edgeencryption.nae.server = url
edgeencryption.nae.port = 9000
edgeencryption.nae.protocol = ssl
edgeencryption.nae.keystore.path = keystore/safenet_truststore
edgeencryption.nae.keystore.password = keystore password
edgeencryption.encrypter.nae.user.1 = nae.user.com
edgeencryption.encrypter.nae.password.1 = <ChangeMe>
```

4. For each encryption key stored in a SafeNet key store, enter the properties for the encryption key.

You will have this set of properties for each encryption key stored in a NAE key store. A number is appended to each property to make the properties unique. For example, `edgeencryption.encrypter.type.3`.

**Table 625: Properties for encryption keys stored in a NAE device**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.encrypter.type</td>
<td>Defines a type of key store system. This property is specified for each key. Value is <code>nae</code>.</td>
</tr>
<tr>
<td>edgeencryption.encrypter.key</td>
<td>Specifies the key name. This property is specified for each key. This name is used to specify the default keys. This is the key alias included as part of the metadata included with each encrypted item, which means that it is stored on the instance.</td>
</tr>
</tbody>
</table>

An example for an encryption key stored in a SafeNet key store.

```plaintext
edgeencryption.encrypter.type.3 = nae
edgeencryption.encrypter.key.3 = naekey128
```

5. Save and close the `edgeencryption.properties` file.

Create and configure encryption keys using Java KeyStore

You can use the keytool shipped with the encryption proxy distribution to create AES 128 and AES 256 encryption keys.

You must use the Java 1.8 version of the keytool utility. A copy of the utility can be found in `<proxy install dir>/java/jre/bin/keytool`.

To find out more about the keytool utility, see the [Java SE Documentation](https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#keytool).
1. Change to the key store directory, `<installation directory>/keystore/`.

2. To create the encryption key, run one of the following commands.

   **Note:** If you choose to run these commands from a directory other than the key store directory, that is you skipped the previous step, you must change the `-keystore` option to include the path from your current directory to the key store directory. For example, if you were in the `<installation directory>/bin` directory, the option would be `-keystore ../keystore/keystore.jceks`.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES 128</td>
<td><code>keytool -genseckey -alias 128bitkey -keyalg aes -keysize 128 -keystore keystore.jceks -storetype jceks</code></td>
</tr>
<tr>
<td>AES 256</td>
<td><code>keytool -genseckey -alias 256bitkey -keyalg aes -keysize 256 -keystore keystore.jceks -storetype jceks</code></td>
</tr>
</tbody>
</table>

3. For each encryption key stored in a Java KeyStore key store, enter the properties for the encryption key.
   a) Change to the `<installation directory>/conf/` directory.
   b) Open the `edgeencryption.properties` file.
   c) Enter the properties for the encryption key.

   You will have this set of properties for each encryption key stored in a Java KeyStore key store. A number is appended to each property to make the properties unique. For example, `edgeencryption.encrypter.type.2`.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.encrypter.type</code></td>
<td>Defines a type of key store system. This property is specified for each key. Value is keystore.</td>
</tr>
<tr>
<td><code>edgeencryption.encrypter.key</code></td>
<td>Specifies the key name. This property is specified for each key. This name is used to specify the default keys. This is the key alias included as part of the metadata included with each encrypted item, which means that it is stored on the instance. The key name must use lowercase letters.</td>
</tr>
<tr>
<td><code>edgeencryption.encrypter.password</code></td>
<td>Specifies the password for accessing the key store.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.https.keystore.path</code></td>
<td>Specifies the URL of the KeyStore.</td>
</tr>
</tbody>
</table>
An example for a Java KeyStore key store.

```plaintext
edgeencryption.encrypter.type.2 = keystore
edgeencryption.encrypter.file.2 = keystore/keystore.jceks
edgeencryption.encrypter.password.2 = password
edgeencryption.encrypter.key.2 = keyalias
```

4. Save and close the `edgeencryption.properties` file.

Create and configure an encryption key stored in a file

You can use a file as a key store. Each file holds a single encryption key.

You must create the encryption key separately. The files must be in the `<installation directory>/conf` directory.

1. Create a file store.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES 128</td>
<td>Place the encryption key, exactly 16 bytes, into the file.</td>
</tr>
<tr>
<td>AES 256</td>
<td>Place the encryption key, exactly 32 bytes, into the file.</td>
</tr>
</tbody>
</table>

2. Update the `edgeencryption.properties` file.
   a) Change to the `<installation directory>/conf/` directory.
   b) Open the `edgeencryption.properties` file.
   c) Enter the properties for the file store.

You will have this set of properties for each encryption key stored in a file. A number is appended to each property to make the properties unique. For example, `edgeencryption.encrypter.type.1`.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.encrypter.type</code></td>
<td>Defines the type of key store system. This property is specified for each key. Value is file.</td>
</tr>
<tr>
<td><code>edgeencryption.encrypter.file</code></td>
<td>Specifies the path and file name of the key's text file on the proxy.</td>
</tr>
<tr>
<td><code>edgeencryption.encrypter.key</code></td>
<td>Specifies the key name. This property is specified for each key. This name is used to specify the default keys. This is the key alias included as part of the metadata included with each encrypted item, which means that it is stored on the instance.</td>
</tr>
</tbody>
</table>

An example for a file system key store.

```plaintext
edgeencryption.encrypter.type = file
edgeencryption.encrypter.file = conf/mykey.txt
```
Configure the default encryption key properties

The encryption key default properties specify the keys that are to be used for the different encryption types. All AES 128 encryption types use the encryption key specified by the `edgeencryption.encrypter.default.key128` property. All AES 256 encryption types use the encryption key specified by the `edgeencryption.encrypter.default.key256` property.

1. Change to the `<installation directory>/conf/` directory.
2. Open the `edgeencryption.properties` file.
3. Enter the encryption key default assignments.

**Table 628: Properties for the encryption keys being used**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.encrypter.default.key128</code></td>
<td>Specifies the name of the current AES 128 key. An AES 128 key must be available even if it is not used. Must be the same for all proxies.</td>
</tr>
<tr>
<td><code>edgeencryption.encrypter.default.key256</code></td>
<td>Specifies the name of the current AES 256 key. Must be the same for all proxies.</td>
</tr>
</tbody>
</table>

**Note:** If you do not plan to use AES 256 encryption, remove this property from the configuration file.

4. Save and close the `edgeencryption.properties` file.

Set up password encryption

You can encrypt passwords in the `edgeencryption.properties` file.

Encrypting passwords in the `edgeencryption.properties` file is good for security, but makes it difficult to debug connection and access issues during initial start up. Make sure that the encryption proxy is set up and running before you set this property. Always set this property in production environments.

1. Change to the `<installation directory>/conf/` directory.
2. In the `conf` directory, create a text file that contains the password you want to use to encrypt the passwords in the `edgeencryption.properties` file.
3. Open the `edgeencryption.properties` file.
4. Enter the password property.
Table 629: Password property

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.encrypter.properties.password</td>
<td>Name of the file in the conf folder that contains the clear text password or phrase used to encrypt passwords in the edgeencryption.properties file. If this property is not set, the passwords are not encrypted. Leave this property blank until after the proxy configuration has been set up and tested.</td>
</tr>
</tbody>
</table>

5. Save and close the `edgeencryption.properties` file.

Set the clear text and IV properties

Set the clear text and IV (initialization vector) properties during the initial installation. Make sure that these properties are the same for all proxies.

1. Change to the `<installation directory>/conf/` directory.
2. Open the `edgeencryption.properties` file.
3. Replace the default values for these proxy properties with values that are unique for your installation.

Table 630: General encryption properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.customer.assigned.known.cleartext</td>
<td>Clear text to let the instance verify that all proxies are using the same keys. At start up, the proxy encrypts the clear text and sends the encrypted text to the instance. The instance does not know the clear text, nor are keys sent to the instance. This property must be the same for all proxies.</td>
</tr>
<tr>
<td>edgeencryption.encrypter.static.iv</td>
<td>The static IV (initialization vector) to be used in equality preserving and order preserving encryption. This property must be the same for all proxies, and it must be exactly 16 bytes (e.g. 16 ASCII characters.)</td>
</tr>
</tbody>
</table>

4. Save and close the `edgeencryption.properties` file.

Lock the proxy configuration

If you want to prevent encryption configuration changes to the proxy in production, set the proxy locked property.

1. Change to the `<installation directory>/conf/` directory.
2. Open the `edgeencryption.properties` file.
3. Set the proxy locked property.
Table 631: Proxy configuration locked property

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.proxy.locked</td>
<td>When true, the proxy does not accept encryption configuration changes or encryption rule changes from the instance. Use this on the production instance after all fields and tables that are to be encrypted have been configured.</td>
</tr>
</tbody>
</table>

4. Save and close the `edgeencryption.properties` file.

Configure the order preserving database properties

If you are using an order preserving encryption type, you must set the Edge Encryption proxy properties for the order preserving database.

1. Change to the `<installation directory>/conf/` directory.
2. Open the `edgeencryption.properties` file.
3. Enter the properties for the order preserving database.

Table 632: Order preserving database properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edgeencryption.db.url</td>
<td>Location of the database used to support order preserving encryption types. Must be the same for all proxies.</td>
</tr>
<tr>
<td>edgeencryption.db.user</td>
<td>User name for accessing the order preserving database. Must be the same for all proxies.</td>
</tr>
<tr>
<td>edgeencryption.db.password</td>
<td>The password for accessing the order preserving database. The default is blank. Must be the same for all proxies.</td>
</tr>
<tr>
<td>edgeencryption.db.name</td>
<td>The name of the database used to support order preserving encryption types. Must be the same for all proxies.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> <code>edgeencryption</code></td>
</tr>
<tr>
<td>edgeencryption.db.bootstrap.file</td>
<td>The bootstrap file for the order preserving database. The file is relative to the <code>sql/</code> directory. Must be the same for all proxies. This parameter should not be changed under normal circumstances.</td>
</tr>
</tbody>
</table>

4. Save and close the `edgeencryption.properties` file.
Set the proxy server initial and upper bound memory limits

The proxy server must have a minimum of 4 GB of RAM available (6 GB recommended). The initial and upper bound memory limits determine how much memory the proxy server can consume.

1. In your proxy server directory, open `<install dir>/conf/wrapper.conf`.
2. Set the proxy server’s initial memory limit.
   a) Locate the line: `#wrapper.java.initmemory=3`.
   b) Replace it with: `wrapper.java.initmemory=6144`.
3. Optionally set the upper bound memory limit.
   Because an upper bound memory limit is not set in the base system, the proxy server may utilize all available memory. If other services are running on the server, you may want to set the upper bound memory limit.
   a) Locate the line: `wrapper.java.maxmemory=<max_memory_specified_in_MB>`.
   b) Set the max memory to the desired number.
4. Save and close the file.

Edge Encryption general properties

The proxy configuration file contains properties that should not be changed under normal circumstances. Generally, these properties should not be changed.

Table 633: Proxy configuration properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.config.poll.interval</code></td>
<td>The poll interval in seconds. The default setting means that it takes 5 seconds for the proxy to learn of encryption configuration changes. Larger values cause the instance to take longer to detect an offline proxy.</td>
</tr>
<tr>
<td><strong>Note:</strong> Do not change this property.</td>
<td></td>
</tr>
<tr>
<td><code>edgeencryption.rules.dir</code></td>
<td>Folder where the encryption rules are stored on the proxy.</td>
</tr>
<tr>
<td><code>edgeencryption.encryption.order_preserving.cache</code></td>
<td>Whether caching is used to support order preserving encryption types.</td>
</tr>
<tr>
<td><code>edgeencryption.encryption.order_preserving.cache</code></td>
<td>Maximum cache size, in bytes.</td>
</tr>
<tr>
<td><code>edgeencryption.jobs.concurrency</code></td>
<td>Maximum number of mass encryption jobs that can run concurrently on this proxy.</td>
</tr>
<tr>
<td><code>edgeencryption.jobs.requests_per_second</code></td>
<td>Number of http job requests per second that can be sent to the instance by this proxy.</td>
</tr>
<tr>
<td><code>edgeencryption.attachments.request.timeout.seconds</code></td>
<td>Attachment upload request timeout in seconds.</td>
</tr>
<tr>
<td><code>edgeencryption.request.buffer.size</code></td>
<td>If an encryption request is larger than this size, the excess is saved to disk. You should not change this number.</td>
</tr>
</tbody>
</table>
### Property Description

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.register.retry.count</code></td>
<td>Maximum number of times the proxy will ping the instance to try to register. The default is 0 (no limit).</td>
</tr>
</tbody>
</table>

---

**Add an additional proxy on Linux**

After the first Edge Encryption proxy is properly configured and tested, you can set up additional proxies on Linux.

**Note:** Make sure that all proxies have the same encryption keys and the same RSA key pair used to digitally sign encryption configuration and encryption rules.

1. Install the proxy using the command for Linux. See *Install the encryption proxy on Linux* on page 2648.
2. Copy all the encryption keys and the `edgeencryption.properties` file from the first proxy to the new proxy.
   
   Encryption keys may be located in the proxy keystore, in the `/keys` directory, or in a SafeNet KeySecure keystore.
3. Open the `edgeencryption.properties` file on the new proxy.
4. Change the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.proxy.name</code></td>
<td>Unique name of the proxy server</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.host</code></td>
<td>The server name, IP address, or fully-qualified domain name of the computer running the proxy. Do not change this property if installing the proxy server on the same machine the properties file was copied from.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.http.port</code></td>
<td>Port on the proxy for HTTP communication. Must be unique across all processes on the machine.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.https.port</code></td>
<td>Port on the proxy for HTTPS communication. Must be unique across processes on the machine.</td>
</tr>
</tbody>
</table>

5. Save and close the file.
6. Launch the proxy using the appropriate command. See *Start the Edge Encryption proxy* on page 2667.

---

**Add an additional proxy on Windows**

After the first Edge Encryption proxy is configured, you can set up additional proxies on Windows.
**Note:** Make sure that all proxies have the same encryption keys and the same RSA key pair used to digitally sign encryption configuration and encryption rules.

1. Install the proxy using the command for Windows. See *Install the encryption proxy on Windows* on page 2649

2. Copy the encryption keys and the `edgeencryption.properties` file from the first proxy to the new proxy.

   Encryption keys may be located in the proxy keystore, in the `/keys` directory, or in a SafeNet KeySecure keystore.

3. Open the `edgeencryption.properties` file on the new proxy and change the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>edgeencryption.proxy.name</code></td>
<td>Unique name of the proxy server.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.host</code></td>
<td>The server name, IP address, or fully qualified domain name of the computer running the proxy. Do not change this property if installing the proxy server on the same machine the properties file was copied from.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.http.port</code></td>
<td>Port on the proxy for HTTP communication. Must be unique across processes on the machine.</td>
</tr>
<tr>
<td><code>edgeencryption.proxy.https.port</code></td>
<td>Port on the proxy for HTTPS communication. Must be unique across processes on the machine.</td>
</tr>
</tbody>
</table>

4. **Caution:** You must perform this step before launching the proxy server.

   Open the `conf/wrapper.conf` file on the new proxy and add the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wrapper.ntservice.name</code></td>
<td>Unique name of the Edge Encryption proxy service.</td>
</tr>
<tr>
<td><code>wrapper.ntservice.displayname</code></td>
<td>Edge Encryption proxy service display name.</td>
</tr>
<tr>
<td><code>wrapper.ntservice.description (Optional)</code></td>
<td>Proxy server description.</td>
</tr>
</tbody>
</table>

5. Save and close the file.

6. Launch the proxy using the appropriate command. See *Start the Edge Encryption proxy* on page 2667.

**Edge Encryption ODBC driver integration**

Configure your ODBC driver to query data encrypted by Edge Encryption. The Edge Encryption proxy server encrypts ODBC driver requests to the ServiceNow instance when Edge Encryption is integrated with the ODBC driver.

Encrypted responses from the instance are decrypted through the Edge Encryption proxy server before passing to the ODBC driver in your network.
For a successful integration, the ODBC driver must trust the Edge Encryption proxy server certificate. If the Edge Encryption proxy server certificate is signed by a Certificate Authority trusted by the ODBC driver, the Edge Encryption proxy server is automatically trusted. However, if a Certificate Authority trusted by the ODBC driver has not signed the Edge Encryption proxy server certificate, you must import the self-signed certificate to the ODBC truststore.
Import a self-signed certificate to the ODBC truststore

If a Certificate Authority trusted by the ODBC driver has not signed the Edge Encryption proxy server certificate, you must import a self-signed certificate to the ODBC truststore. You can export the certificate from the Edge Encryption proxy server and import it into the ODBC truststore.

To determine whether a Certificate Authority trusted by the ODBC driver has signed the Edge Encryption proxy server certificate, run the following command in the keystore directory in the proxy home directory to view a list of Certificate Authorities trusted by the ODBC driver:

```
keytool -keystore "<ODBC directory>\ip\Java\jre\lib\security\cacerts" -list
```

1. Change to the keystore directory in the proxy home directory.
2. Check the keystore for the self-signed certificate.
   a) To check the keystore for the certificate, you can run the following command to list all the items in the keystore.
      ```
      keytool -list -keystore keystore.jceks -storetype jceks -v
      ```
   b) Locate the key alias in the list of items.
3. Using the key alias, export the certificate to a .cer file.
   ```
   keytool -export -alias <key alias> -keystore keystore.jceks -storetype jceks -rfc -file <file name>.cer
   ```
4. Change to your ODBC truststore directory: `ODBC\ip\Java\jre\lib\security\cacerts`.
5. Import the certificate to your ODBC truststore.
   ```
   keytool -keystore cacerts -importcert -alias $<key alias> -file <file name>.cer
   ```

Set the ODBC driver properties

Set the ODBC driver properties to route requests through the Edge Encryption proxy server.

1. In Windows, navigate to Start > Programs > ServiceNow ODBC Management Console.
2. Expand the console tree root to: ServiceNow ODBC Manager\Manager\<installation location>\Services\ServiceNow_ODBC\Data Source Settings\ServiceNow\IP Parameters.
3. Double-click the DataSourceIPProperties attribute.
4. Change the Value to the URL of your Edge Encryption proxy server, such as https://<IP address>:<port>
5. Click OK.

The ODBC driver is now configured to route requests to the instance through the Edge Encryption proxy server.

**Edge Encryption MID Server integration**

Configure the MID Server to route data through an Edge Encryption proxy server.

When integrated with the MID Server, the Edge Encryption proxy server acts as the MID Server's endpoint. The Edge Encryption proxy server then encrypts and decrypts data passing between the ServiceNow instance and the MID Server.

**Limitations when integrating with the MID Server**

When MID Server data is configured to pass through the Edge Encryption proxy server, the following limitations apply:

- While string fields in an ECC Queue record can be encrypted, the Edge Encryption proxy server cannot encrypt values within the structured data in the ECC Queue payload field. Alternatively, the entire ECC Queue payload field can be encrypted, or the MID Server can use REST calls or other MID Server capabilities to pass values marked for encryption to the Edge Encryption proxy server.
- Encrypted data cannot be used with Discovery or Service Mapping.

**Point the MID Server to the Edge Encryption proxy server**

To pass data from the MID Server through the Edge Encryption proxy server, update the MID Server configuration file to point the MID Server to the Edge Encryption proxy server.

When configuring the MID Server to pass through the Edge Encryption proxy server, you cannot use the web proxy properties in the MID Server configuration file to route traffic through the Edge Encryption proxy server to your instance. Instead, you must set the Edge Encryption proxy server as the MID Server's endpoint.

1. Navigate to your local MID Server directory and open the `config.xml` file.
2. Find the element `<parameter name="url" value="https://YOUR_INSTANCE.service-now.com" />` and change the value property to the URL of your Edge Encryption proxy server. For example, `http://hostname.mycompany.com:8081`.
   - This step directs the MID Server to pass traffic to the Edge Encryption proxy server instead of the instance. The Edge Encryption proxy server in turn encrypts any necessary fields and passes the payload to the instance.
3. Save and close the file.
4. If running, restart the MID Server.

**Start the Edge Encryption proxy**

After an Edge Encryption proxy is installed and configured, you can start the proxy from the command line.

Before starting the encryption proxy, verify the following:

- The Edge Encryption plugin is activated on the instance.
- The `edgeencryption.properties` file on this machine has been configured.
- If using an order preserving encryption type, the order preserving database is running.
Note: The first time you set up the `edgeencryption.properties` file or change properties, you may want not to set the password encryption property. After you have verified that everything is working, you can set the password encryption property, shut down the proxy, and then restart the proxy.

1. Run the proxy server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| On a Linux machine | 1. cd to ServerName_port  
                     2. Execute ./startup.sh |
| On a Windows machine | 1. cd to ServerName_port/bin  
                        2. Execute edgeencryption.bat install  
                        3. Execute edgeencryption.bat start |

2. Check the log on the proxy server to verify that the proxy is running.

Stop the Edge Encryption proxy

You can stop an Edge Encryption proxy from the command line.

1. Stop the proxy server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On a Linux machine</td>
<td>Execute ./shutdown.sh</td>
</tr>
<tr>
<td>On a Windows machine</td>
<td>Execute edgeencryption.bat stop</td>
</tr>
</tbody>
</table>

2. Check the log on the proxy server to verify the proxy has stopped.

Configure Edge Encryption on the instance

Configure Edge Encryption by configuring fields and attachments for encryption.

Before you encrypt a field, determine what system features might be impacted. Examine all scripts for use of the field.

Before marking a field as encrypted, make any desired adjustments to the field's size. After a field has been configured for encryption, the field size cannot be changed.

Marking a field to be encrypted expands the field size to hold the extra space needed to store the encrypted data. The process of expanding the field size can take a long time depending on how many records the table has.

Test all changes on a sub-production instance before making the changes to the production instance.

To configure field encryption, you must be connected to the instance through the Edge Encryption proxy.

Configure field encryption

Select the fields to be encrypted and identify the encryption type.

Role required: security-admin
Warning: Specifying a field as encrypted cannot be undone. Clicking Submit creates an encryption record that cannot be deleted. Encrypted fields can be marked as inactive (by clearing the Active box), and then the field contents decrypted, but the encryption record cannot be deleted, and the encryption type cannot be changed.

2. Fill in the fields on the form, as appropriate.

Table 634: Edge Encryption configuration

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>The table to be encrypted.</td>
</tr>
<tr>
<td>Type</td>
<td>Whether to encrypt a table column or attachments for the table.</td>
</tr>
<tr>
<td>Column</td>
<td>The table field to be encrypted.</td>
</tr>
<tr>
<td></td>
<td>This field appears when the Type is Column.</td>
</tr>
<tr>
<td>Encryption type</td>
<td>The encryption type to use.</td>
</tr>
</tbody>
</table>

3. Click Submit.

After the encryption record has been added, you can create an encryption job to encrypt existing data. If you do not run an encryption job, the existing data is encrypted the next time it is changed.

Update the Edge Encryption proxy

You must update the Edge Encryption proxy when new versions of the proxy and supporting software become available.

You must manually update each Edge Encryption proxy with the latest security updates for Java. This process is not automatic.

Uninstall the Edge Encryption proxy on Linux

Before installing a new version of the Edge Encryption proxy, you must shut down and uninstall the current version.

You must have access to the computer running the Edge Encryption proxy.

Before shutting down the Edge Encryption proxy, ensure that no users are connected to the instance using the proxy.

The encryption proxy running on Linux operates as a single process. You can end this process to accommodate such tasks as redeploying the encryption proxy to another host machine, updating the proxy version, updating the Java version, or changing the unique name of the encryption proxy when deploying the encryption proxy on multiple proxy servers.

1. You may want to save the edgeencryption.properties file before deleting the distribution directory.
2. Execute the shutdown.sh shell script.
3. Check the log on the proxy server to verify that the proxy server is shutdown.
4. Delete the files in the distribution folder.
Uninstall the Edge Encryption proxy on Windows

Before installing a new version of the Edge Encryption proxy, you must shut down and uninstall the current version.

You must have access to the computer running the Edge Encryption proxy.

Before shutting down the Edge Encryption proxy, ensure that no users are connected to the instance using the proxy.

1. You may want to save the `edgeencryption.properties` file before deleting the distribution directory.
2. Execute `edgeencryption.bat stop`
3. Execute `edgeencryption.bat remove`
4. Check the log on the proxy server to verify that the proxy server is shutdown.
5. Delete the files in the distribution folder.

Rotate encryption keys

Edge Encryption provides the tools to support encryption key rotation.

Before setting an encryption key as the default key on any proxy, add the encryption key to all proxies and restart the proxies. This ensures that the proxies have the key to decrypt fields that are encrypted with the new default key assignment. You must restart the proxy after adding a new key and after setting a new key as the default key.

Note: Before removing a key from the proxy configuration files and the key store, ensure that no data on the instance uses the key. You can do this by setting up and running a single key rotation job.

1. Obtain the new encryption key and make it accessible to all encryption proxies.
2. For each encryption proxy, add the new encryption key.
   a) Edit the encryption properties file to add the new key.
      This can be adding an encryption key stored in a file store, Java KeyStore, or NAE key store and adding the necessary properties. After updating the first proxy, you may be able to copy and paste the new encryption properties into subsequent encryption-proxy properties files.
   b) Shut down and restart the proxy.
3. For each encryption proxy, set the new encryption key as the default key.
   a) Edit the encryption properties file to set the new key as the default key.
   b) Shut down and restart the proxy.
4. If desired, create and run a mass encryption job to encrypt existing data using the new encryption key.
   You can change the default encryption key assignment so that all new data is encrypted using the new key, and let existing data remain encrypted with the old key until the data is accessed again.

Scheduled encryption jobs

If you have the security-admin role, you can schedule several different types of jobs to be performed by the Edge Encryption proxy.

You can schedule jobs for the following:
• Encrypting, decrypting, or re-encrypting data in a table field or attachments for a table.
• Repairing order tokens.
• Recreating the order token database.

Running any of these jobs can be a time-consuming operation which might impact the performance of the Edge Encryption proxy. Schedule these jobs at a time when no users or a minimum set of users are using the system, such as midnight on the weekend.

Schedule an encryption job

You can schedule a job to find and encrypt any unencrypted data in a specified field, using the default encryption key configured for the field. If you do not create an encryption job after configuring a field for encryption, the records are encrypted as they are saved to the instance.

Role required: security-admin

1. Navigate to Edge Encryption Configuration > Encryption Configurations > All.
2. Click the field that you want to schedule an encryption job for.

   The Scheduled Encryption Job form is shown with all fields populated. The bottom of the form shows records for any previous job executions.

4. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>Active</td>
<td>Clear this check box if you want to deactivate this job.</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select Encryption.</td>
</tr>
<tr>
<td>Table</td>
<td>Select a table.</td>
</tr>
<tr>
<td>Column</td>
<td>Select a column.</td>
</tr>
<tr>
<td>Run</td>
<td>Select the period between job executions.</td>
</tr>
<tr>
<td>Starting</td>
<td>Enter the date and time to run the job for the first time.</td>
</tr>
</tbody>
</table>

5. Click the menu icon in the form header and select Save.
6. To see an estimated count of records to be updated, click Estimate Record Count.
7. To run the job immediately, click Execute Now.

Schedule an attachment encryption job

You can schedule a job to find and encrypt any unencrypted attachments for a specified table, using the default encryption key configured for the table.

Role required: security-admin

1. Navigate to Edge Encryption Configuration > Encryption Configurations > All.
2. Click the table you want to schedule an encryption job for.

   The Scheduled Encryption Job form is shown with all fields populated. The bottom of the form shows records for previous job executions.

4. Fill in the fields on the form, as appropriate.
Schedule a decryption job

You can schedule a job to decrypt data in an encrypted field, to store clear data in the instance.

Note: You must mark the encryption record for the field as inactive (clear the Active box) before the decryption job runs, otherwise, nothing happens.

Role required: security-admin

1. Navigate to Edge Encryption Configuration > Encryption Configurations > All.
2. Click the field that you want to decrypt.
   
   The Scheduled Encryption Job form is shown with all fields populated. The bottom of the form shows records for previous job executions.
4. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select Decryption.</td>
</tr>
<tr>
<td>Active</td>
<td>Clear this check box if you want to deactivate this job.</td>
</tr>
<tr>
<td>Table</td>
<td>Select a table.</td>
</tr>
<tr>
<td>Column</td>
<td>Select a column.</td>
</tr>
<tr>
<td>Run</td>
<td>Select the period between job executions.</td>
</tr>
<tr>
<td>Starting</td>
<td>Enter the date and time to run the job for the first time.</td>
</tr>
</tbody>
</table>

5. Click the menu icon in the form header and select Save.
6. To see an estimated count of records to be updated, click Estimate Record Count.
7. To run the job immediately, click Execute Now.
Schedule an attachment decryption job

You can schedule a job to decrypt any encrypted attachments for a specified table, to store clear attachments in the instance.

Note: You must mark the encryption record for the table as inactive (clear the Active box) before the decryption job runs, otherwise, nothing happens.

Role required: security-admin

1. Navigate to Edge Encryption Configuration > Encryption Configurations > All.
2. Click the table with the attachments that you want to decrypt.

The Scheduled Encryption Job form is shown with all fields populated. The bottom of the form shows records for previous job executions.
4. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select Attachment Decryption.</td>
</tr>
<tr>
<td>Active</td>
<td>Clear the check mark if you want to deactivate this job.</td>
</tr>
<tr>
<td>Table</td>
<td>Select a table.</td>
</tr>
<tr>
<td>Run</td>
<td>Select the period between job executions.</td>
</tr>
<tr>
<td>Starting</td>
<td>Enter the date and time to run the job for the first time.</td>
</tr>
</tbody>
</table>

5. Click the menu icon in the form header and select Save.
6. To see an estimated count of records to be updated, click Estimate Record Count.
7. To run the job immediately, click Execute Now.

Schedule a mass key rotation job

You can schedule a job to find data encrypted with old keys and then re-encrypt the data with the current default encryption keys. The data is decrypted before it is re-encrypted with the current default key.

Role required: security-admin

1. Navigate to Edge Encryption Configuration > Maintenance > Schedule Mass Key Rotation.
2. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select Mass Key Rotation.</td>
</tr>
<tr>
<td>Active</td>
<td>Clear this check box if you want to deactivate this job.</td>
</tr>
<tr>
<td>Run</td>
<td>Select the period between job executions.</td>
</tr>
</tbody>
</table>
Schedule a single key rotation job

You can schedule a job to find data encrypted using a specified key alias and then re-encrypt the data with the current default encryption key. The data is decrypted before it is re-encrypted with the default key.

Role required: security-admin

Before scheduling this job, make sure you update the default key in the edgeencryption.properties files of all Edge Encryption proxies.

1. Navigate to Edge Encryption Configuration > Maintenance > Schedule Single Key Rotation.
2. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select Single Key Rotation.</td>
</tr>
<tr>
<td>Key Alias</td>
<td>Enter the key to be retired. Make sure this key is no longer the default key in any edgeencryption.properties files.</td>
</tr>
<tr>
<td>Active</td>
<td>Clear this check box if you want to deactivate this job.</td>
</tr>
<tr>
<td>Run</td>
<td>Select the period between job executions.</td>
</tr>
<tr>
<td>Starting</td>
<td>Enter the date and time to run the job for the first time.</td>
</tr>
</tbody>
</table>

3. Click the menu icon in the form header and select Save.
4. To see an estimated count of records to be updated, click Estimate Record Count.

Schedule an order token repair job

You can schedule a job to find and repair fields where the order token is missing.

Role required: security-admin

Use these jobs to repair individual fields in a table or to repair all fields using order preserving encryption. Run this job when the order token database has been offline while the instance has been running, which results in order preserving fields that are missing order tokens.

1. Navigate to Edge Encryption Configuration > Maintenance > Schedule Order Token Repair.
2. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name.</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select Order Token Repair.</td>
</tr>
</tbody>
</table>
Schedule a database recovery job

Run this job when the database on the client system that supports order preserving encryption has lost data. This job finds all records that have been encrypted with a token (order preserving encryption type) and sends them to the proxy so that the order preserving database can be rebuilt.

Role required: security-admin

1. Navigate to **Edge Encryption Configuration > Maintenance > Schedule Database Recovery**.
2. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for this job.</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select <strong>Database Recovery</strong>.</td>
</tr>
<tr>
<td>Active</td>
<td>Clear this check box if you want to deactivate this job.</td>
</tr>
<tr>
<td>Run</td>
<td>Select the period between job executions.</td>
</tr>
<tr>
<td>Starting</td>
<td>Enter the date and time to run this job for the first time.</td>
</tr>
</tbody>
</table>

3. Click the menu icon in the form header and select **Save**.
4. To see an estimated count of records to be updated, click **Estimated Record Count**.

Edge Encryption monitoring

You can monitor sessions that use Edge Encryption proxies.

The instance tracks all encryption proxies. Each encryption proxy registers when it starts up. The instance is notified when:

- A new encryption proxy starts up.
- An encryption proxy is intentionally shut down.

If an encryption proxy attempts to register with an instance that does not have Edge Encryption installed, the proxy will not start.
All encryption configuration files are audited. Deleted records are audited on all encryption configuration files. Audit records are put in the sys_audit table. To view the history of a specific configuration record, view the record, and click **History > List** in the menu. The Mass Encryption Job file is not audited.

You can also monitor the following lists and file for information about your proxies:

| Edge Encryption Configuration > Diagnostics And Troubleshooting > Invalid Insert Attempts | List of attempts to save the following data to encrypted fields:
| --- | --- |
|  | • Unencrypted data.
|  | • Data that did not come from an Edge Encryption proxy.
|  | The instance rejects and then logs any attempts to save this data. If you have the security-admin role, you can view the logs in the Invalid Insert Attempts list.

| Edge Encryption Configuration > Proxies > Online | List of online proxies.
| --- | --- |

| Edge Encryption Configuration > Proxies > All | List of all proxies.
| --- | --- |

| System log | The instance periodically checks for messages from each encryption proxy, and it logs an error if a proxy has not sent a message in the required time frame. The log message contains information about the encryption proxy and the last time the proxy pinged the instance. If the instance determines that none of the encryption proxies are online, it logs a message. These messages are added to the system log.

| xmlstats | Information about the number of registered encryption proxies, and the number of pending and running mass encryption jobs.

## Edge Encryption logging

Edge Encryption logs information on the instance and on each proxy server.

You can view a log file listing all cases of when an attempt was made to save unencrypted data to an encrypted field. Navigate to **Edge Encryption Configuration > Diagnostics And Troubleshooting > Invalid Insert Attempts**. An administrator can trace back to why data was not saved.

There are encryption log files on each proxy server.

## Encryption rules

It may be necessary to write encryption rules when you want to encrypt data passed as part of GET/POST requests to processors or APIs on the instance. You can create rules for mapping elements of fields in requests to Glide table-field names.

Edge Encryption uses encryption rules executed on the proxy to map fields in a HTTP request to fields in a table. Each encryption rule is composed of a condition and an action. The condition identifies the type of request, and the action performs the mapping from fields in the request to fields in a table.

A set of encryption rules is installed as part of the Edge Encryption plugin. These rules handle the core platform use cases such as editing a field from the list edit form, updating a record from record form, direct
web services, and the REST API. Applications created using standard forms and lists should work without needing custom encryption rules.

You must write encryption rules for scripted processors, scripted web services, scripted REST APIs, UIs, or Ajax scripts you develop, if data in the requests needs to be encrypted.

You must have the security-admin role to be able to create a rule. The script is checked for compliance with JavaScript syntax before the rule is saved.

Except for attachment requests, each HTTP request goes through the process of executing the encryption rule conditions until either all conditions return false, or one condition returns true. When a condition returns true, the action part of the rule is run, and the result is forwarded to the instance, i.e. no other conditions are evaluated. As a result, encryption rule conditions must be as specific as possible to avoid inadvertently satisfying the condition. If a generic condition for a rule is unavoidable, the rule should be marked with a high order value so that more specific rules are evaluated first.

Encryption rules are written using a combination of JavaScript and the Edge Encryption API that allows you to easily iterate through post parameters, and JSON and XML content in the request body. The API uses expressions similar to XPath to navigate through both JSON and XML content.

The API uses stream parsing to parse JSON and XML data, so it is recommended that operations on the data in the action part of an encryption rule process the data in one pass. Trying to fetch and parse the content of the request body multiple times may lead to unexpected results.

When creating encryption rules, you cannot use Glide APIs, script includes, business rules, or any global parameters such as `current`. Since the rules are created for HTTP Post and HTTP Get objects, a global variable `request` is available.

When creating encryption rules, you cannot use APIs from the white list manager or scoped applications.

Encryption rule conditions

An encryption rule condition should return true if the rule is to handle the HTTP request; otherwise, it should return false. The condition can use the method type, content type, URL path, or any URL query string parameters to determine if the rule should handle the request.

The condition has access to these fields via the `request` object.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>path</td>
<td>The path portion of the URL</td>
</tr>
<tr>
<td>requestMethod</td>
<td>GET or POST</td>
</tr>
<tr>
<td>contentType</td>
<td>The Content-Type header field</td>
</tr>
<tr>
<td>urlParams</td>
<td>The parameters in the query string. This can also be evaluated to a String.</td>
</tr>
<tr>
<td>postParams</td>
<td>If this is a form post, this contains the post parameters</td>
</tr>
</tbody>
</table>

An example:

```javascript
function SampleCondition(request) {
  if (endsWith(request.path, "/sample_processor.do")) {
    return true;
  }
  return false;
```
An example accessing a URL query string parameter:

```javascript
//Check if the property ‘myParam’ exists in the urlParams object.
if (request.urlParams.myParam) {
  var myParam = request.urlParams.myParam;
}
```

Encryption Rule Actions

An encryption-rule action is responsible for mapping fields in an HTTP request to fields in a table. In addition to the fields available in the condition part of a rule, the action part of a rule has access to the JSON or XML content in the request.

**POST parameters**

To access POST parameters in any part of an encryption rule, you get the `postParams` property from the `request` object. Each POST parameter is a property of the `postParams` object.

An example of accessing the `postParams` object in a string context. The parameter evaluates to the string representation of the value.

```javascript
if (request.postParams.myparam == "hello") {
  var paramValue = request.postParams.myparam;
}
```

An example accessing the `postParams` object in a `for` loop.

```javascript
for (var paramName in request.postParams) {
  var paramValue = request.postParams[paramName];
}
```

**XML content**

You can access XML data from either a post parameter or from the HTTP request content.

You access XML data in post parameter by calling `request.postParams.<parameter name>.getAsXmlContent()`.

You access XML data in the request content by calling `request.getAsXmlContent()`.

Calling either of these two functions will return an object of type `XMLContent` that provides methods to iterate over the XML content.

An example using an `XMLContent` object with the following XML.

```xml
<records>
  <record>
    <name>Test Record 1</name>
    <description>Test Record 1 Description</description>
  </record>
  <record>
    <name>Test Record 2</name>
    <description>Test Record 2 Description</description>
  </record>
</records>
```

To iterate through the XML, you can do the following.

```javascript
var xmlContent = request.getAsXmlContent();
// This outer loop will iterate over all record elements
```
for (var xmlElementIter = xmlContent.getIterator('/records/record');
xmlElementIter.hasNext(); ) {
    var xmlElement = xmlElementIter.next();
    // This inner loop will iterate over all elements within each record
    element
    for (var childIter = xmlElement.getIteratorOverAllChildren();
    childIter.hasNext(); ) {
        var childElement = childIter.next();
        var fieldname = childElement.getName();
    }
}

By calling `xmlContent.getIterator('/records/record')` you can iterate over the record elements by calling `hasNext()` to determine if there is another element with the path specified in the expression argument in the call to `getIterator()`. The `next()` method gives you the next element found matching the expression. An object of type `XMLElement` is returned.

**JSON content**

You can access JSON data from either a post parameter or from the HTTP request content.

You access JSON content from a post parameter by calling `request.postParms.<parameter name>.getAsJsonContent()`.

You access JSON content from the HTTP request by calling `request.getAsJsonContent()`.

Calling either of the methods returns an object of type `JsonNode`. With a `JsonNode` object, you can easily iterate over the properties by calling `iterator()` with an XPath-like expression.

An example using a `JsonNode` object with the following value.

```javascript
{
    records: [
        {
            name: 'Test Record 1',
            description: 'Test Record 1 Description'
        },
        {
            name: 'Test Record 1',
            description: 'Test Record 1 Description'
        }
    ]
}
```

To iterate through the JSON, you can do the following.

```javascript
var jsonContent = request.getAsJsonContent();
//Get the table name from a URL parameter
var tableName = request.urlParams.tableName;

// This outer loop iterates over all elements in the records array
for (var jsonNodeIter = jsonContent.getIterator('/records');
jsonNodeIter.hasNext(); ) {
    var jsonNode = jsonNodeIter.next();
    // This inner loop iterates over all child elements in an object in the
    records array
    for (var propertyIter = jsonNode.iterator(); propertyIter.hasNext(); ) {
        var property = propertyIter.next();
        var propertyName = property.getName();

        //The valueFor method replaces the value of the 'current' element
        with an encrypted
        //value if the field in the table provided as arguments is marked as
        encrypted. Otherwise,
        //it leaves the value as-is.
```
Error handling
When a rule throws an exception the proxy disables the rule.
If an encryption rule condition or action throws an exception, the exception is caught and the encryption rule is disabled. The proxy continues to evaluate the remaining rules in order as if the rule that threw the exception was not present. The exception is logged in the localhost log with a stack trace identifying the root cause of the problem.

request API
The request object is a global object available in Edge Encryption rule action scripts.

request - getEncodedQueryFor(String tableName)
Specifies that the value of the parameter is an encoded query for the specified table.
Calling this function on a parameter tells the proxy that the value of the parameter represents an encoded query for the specified table. The proxy parses the encoded query and encrypts the values for fields in the encoded query that need to be encrypted.

This method is available only in an Edge Encryption rule action script.

Table 636: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>A table name</td>
</tr>
</tbody>
</table>

Table 637: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

This example is from the List Request encryption rule.

```javascript
function ListRequestAction(request) {
    var path = request.path;
    var tableNameEndIndex = path.indexOf("_list.do");
    var tableName = path.substring(1, tableNameEndIndex);
    request.urlParams.sysparm_query.encodedQueryFor(tableName);
}
```

request - valueFor(String tableName, String fieldName)
Identifies that the parameter maps to the specified field in the specified table.
Calling this function on a parameter value tells the proxy that the value for this parameter maps to the specified field in the specified table. The proxy then decides if the field needs to be encrypted.

This method is available only in an Edge Encryption rule action script.
Table 638: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>The table name</td>
</tr>
<tr>
<td>fieldName</td>
<td>String</td>
<td>The field name</td>
</tr>
</tbody>
</table>

Table 639: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

This example is from the Form Post encryption rule.

```javascript
function FormPostAction(request) {
    var tableName = request.postParams.sys_target;
    for (var paramName in request.postParams) {
        var fieldName = paramName.replace("sys_display.original." + "", "");
        fieldName = fieldName.replace("sys_original." + "", "");
        fieldName = fieldName.replace("sys_display." + "", "");
        fieldName = fieldName.replace(tableName + ".", "");
        request.postParams[paramName].valueFor(tableName, fieldName);
    }
}
```

request - getAsJsonContent()

Returns the request as an object of type `JsonNode`.

This method is available only in an Edge Encryption rule action script.

Table 640: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 641: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JsonNode</td>
<td>The request</td>
</tr>
</tbody>
</table>

This example is from the REST Json encryption rule.

```javascript
function RESTJSONAction(request) {
    var urlArray = request.path.split('/');
    var tableName = urlArray[urlArray.length - 1];

    if (request.urlParams && request.urlParams.sysparm_query) {
        request.urlParams.sysparm_query.encodedQueryFor(tableName);
    }
}
```
request - getAsXmlContent()

Returns the request content as an object of type **XMLContent**.

This method is available only in an Edge Encryption rule action script.

**Table 642: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 643: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLContent</td>
<td>The request</td>
</tr>
</tbody>
</table>

This example is from the List Edit encryption rule.

```javascript
function ListEditAction(request) {
  var xmlContent = request.postParams.sysparm_xml.getAsXmlContent();
  for (var xmlElementIter = xmlContent.getIterator('record_update');
       xmlElementIter.hasNext(); ) {
    var xmlElement = xmlElementIter.next();
    var tableName = xmlElement.getAttributeValue('table');
    for (var xmlRecordIter = xmlElement.getIterator('record');
         xmlRecordIter.hasNext(); ) {
      var xmlRecord = xmlRecordIter.next();
      for (var xmlFieldIter = xmlRecord.getIterator('field');
            xmlFieldIter.hasNext(); ) {
        var xmlField = xmlFieldIter.next();
        var fieldName = xmlField.getAttributeValue('name');
        for (var childIter = xmlField.getIteratorOverAllChildren();
             childIter.hasNext(); ) {
          var childElement = childIter.next();
          childElement.valueFor(tableName, fieldName);
        }
      }
    }
  }
}
```
The JsonNode class is available in Edge Encryption rule action scripts.

You get a JsonNode object by calling the next() method of a JsonNodeIterator object.

Note: This method can only be used on the root node, but can be used to traverse deep into the JSON object. Subsequent traversals must use the iterator() method.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xPath</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JsonNodeIterator</td>
<td>An object that can iterate over nodes in the JSON object.</td>
</tr>
</tbody>
</table>

This example is from the REST JSON encryption rule.

```javascript
function RESTJSONAction(request) {
  var urlArray = request.path.split('/');
  var tableName = urlArray[urlArray.length - 1];
}
if (request.urlParams & request.urlParams.sysparm_query) {
    request.urlParams.sysparm_query.encodedQueryFor(tableName);
}

var jsonContent = request.getAsJsonContent();
for (var jsonElementItr = jsonContent.iterator(); jsonElementItr.hasNext();) {
    var jsonElement = jsonElementItr.next();
    jsonElement.valueFor(tableName, jsonElement.getName());
}

**JsonNode - getAsString(String propertyName)**

*Returns the string value of the specified property.*

This method is available only in an Edge Encryption rule action script.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>propertyName</td>
<td>String</td>
<td>Name of the property</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The property value</td>
</tr>
</tbody>
</table>

**JsonNode - getName()**

*Returns the name of the current JSON node.*

This method is available only in an Edge Encryption rule action script.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Name of the JSON node</td>
</tr>
</tbody>
</table>
JsonNode - `valueFor(String tableName, String fieldName)`

Specifies that the JSON property maps to the specified field in the specified table.

Calling this method on a JSON property tells the proxy that the value for this property maps to the specified field in the specified table. The proxy then decides if the field needs to be encrypted.

This method is available only in an Edge Encryption rule action script.

### Table 652: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>The table name</td>
</tr>
<tr>
<td>fieldName</td>
<td>String</td>
<td>The field name</td>
</tr>
</tbody>
</table>

### Table 653: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

This example is from the REST JSON encryption rule.

```javascript
function RESTJSONAction(request) {
    var urlArray = request.path.split('/');
    var tableName = urlArray[urlArray.length - 1];

    if (request.urlParams && request.urlParams.sysparm_query) {
        request.urlParams.sysparm_query.encodedQueryFor(tableName);
    }

    var jsonContent = request.getAsJsonContent();
    for (var jsonElementItr = jsonContent.iterator();
         jsonElementItr.hasNext(); ) {
        var jsonElement = jsonElementItr.next();
        jsonElement.valueFor(tableName, jsonElement.getName());
    }
}
```

JsonNode - `encodedQueryFor(String tableName)`

Specifies that the value of the JSON property is an encoded query for the specified table.

Calling this function on an JSON node tells the proxy that the value is an encoded query for the specified table. The proxy parses the encoded query and encrypts the values for fields in the encoded query that need to be encrypted.

### Table 654: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>A table name</td>
</tr>
</tbody>
</table>
This example is from the REST JSON encryption rule.

```javascript
function RESTJSONAction(request) {
  var urlArray = request.path.split('/');
  var tableName = urlArray[urlArray.length - 1];

  if (request.urlParams && request.urlParams.sysparm_query) {
    request.urlParams.sysparm_query.encodedQueryFor(tableName);
  }

  var jsonContent = request.getAsJsonContent();
  for (var jsonElementItr = jsonContent.iterator();
   jsonElementItr.hasNext(); ) {
    var jsonElement = jsonElementItr.next();
    jsonElement.valueFor(tableName, jsonElement.getName());
  }
}
```

**JsonNodeIterator API**

The `JsonNodeIterator` class is available in Edge Encryption rule action scripts.

You get an `JsonNodeIterator` object by calling the `getIterator()` or `iterator()` methods of the `JsonNode` class.

**JsonNodeIterator - hasNext()**

Determines if there is another property available.

This method is available only in an Edge Encryption rule action script.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>True if another property is available.</td>
</tr>
</tbody>
</table>

This example is from the REST JSON encryption rule.

```javascript
function RESTJSONAction(request) {
  var urlArray = request.path.split('/');
  var tableName = urlArray[urlArray.length - 1];

  if (request.urlParams && request.urlParams.sysparm_query) {
    request.urlParams.sysparm_query.encodedQueryFor(tableName);
  }
```
JsonNodeIterator - next()

Returns the next property in the iterator.
This method is available only in an Edge Encryption rule action script.

Table 658: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 659: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JsonNode</td>
<td>The next JsonNode</td>
</tr>
</tbody>
</table>

This example is from the REST JSON encryption rule.

```javascript
function RESTJSONAction(request) {
  var urlArray = request.path.split('/');
  var tableName = urlArray[urlArray.length - 1];

  if (request.urlParams && request.urlParams.sysparm_query) {
    request.urlParams.sysparm_query.encodedQueryFor(tableName);
  }

  var jsonContent = request.getAsJsonContent();
  for (var jsonElementItr = jsonContent.iterator();
       jsonElementItr.hasNext(); ) {
    var jsonElement = jsonElementItr.next();
    jsonElement.valueFor(tableName, jsonElement.getName());
  }
}
```

**XMLContent API**
The `XMLContent` class is a global object available in Edge Encryption rule action scripts.

`XMLContent - getIterator()`

Returns an `XMLElementIterator` object for the XML content.
This method is available only in an Edge Encryption rule action script.
### Table 660: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 661: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLElementIterator</td>
<td>An object that will iterate over elements in the XMLContent object.</td>
</tr>
</tbody>
</table>

This example is from the List Edit encryption rule.

```javascript
function ListEditAction(request) {
    var xmlContent = request.postParams.sysparm_xml.getAsXmlContent();
    for (var xmlElementIter = xmlContent.getIterator('record_update');
        xmlElementIter.hasNext(); ) {
        var xmlElement = xmlElementIter.next();
        var tableName = xmlElement.getAttributeValue('table');
        for (var xmlRecordIter = xmlElement.getIterator('record');
            xmlRecordIter.hasNext(); ) {
            var xmlRecord = xmlRecordIter.next();
            var fieldName = xmlRecord.getAttributeValue('field');
            for (var xmlFieldIter = xmlRecord.getIterator('field');
                xmlFieldIter.hasNext(); ) {
                var xmlField = xmlFieldIter.next();
                var childIter = xmlField.getIteratorOverAllChildren();
                var childElement = childIter.next();
                childElement.valueFor(tableName, fieldName);
            }
        }
    }
}
```

**XMLContent - getIterator(String xPath)**

Returns an XMLElementIterator object for the XML content based on the specified parameter.

This method is available only in an Edge Encryption rule action script.

### Table 662: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xPath</td>
<td>String</td>
<td>An XPath like expression that specifies where in the XMLContent object to start.</td>
</tr>
</tbody>
</table>
### Table 663: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLElementIterator</td>
<td>An object that will iterate over elements in the XMLContent object.</td>
</tr>
</tbody>
</table>

This example is from the List Edit encryption rule.

```javascript
function ListEditAction(request) {
    var xmlContent = request.postParams.sysparm_xml.getAsXmlContent();
    for (var xmlElementIter = xmlContent.getIterator('record_update');
        xmlElementIter.hasNext(); ) {
        var xmlElement = xmlElementIter.next();
        var tableName = xmlElement.getAttributeValue('table');
        for (var xmlRecordIter = xmlElement.getIterator('record');
            xmlRecordIter.hasNext(); ) {
            var xmlRecord = xmlRecordIter.next();
            var xmlFieldIter = xmlRecord.getIterator('field');
            xmlFieldIter.hasNext(); } {
            var xmlField = xmlFieldIter.next();
            var fieldName = xmlField.getAttributeValue('name');
            for (var childIter = xmlField.getIteratorOverAllChildren();
                childIter.hasNext(); ) {
                var childElement = childIter.next();
                childElement.valueFor(tableName, fieldName);
            }
        }
    }
}
```

**XMLElementIterator API**

The `XMLElementIterator` class is available in Edge Encryption rule action scripts.

You get an `XMLElementIterator` object by calling the `getIterator()` method of the `XMLContent` class.

**XMLElementIterator - hasNext()**

Determines if there is another element available.

This method is available only in an Edge Encryption rule action script.

### Table 664: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 665: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>True if another element is available.</td>
</tr>
</tbody>
</table>
This example is from the List Edit encryption rule.

```javascript
function ListEditAction(request) {
  var xmlContent = request.postParams.sysparm_xml.getAsXmlContent();
  for (var xmlElementIter = xmlContent.getIterator('record_update');
      xmlElementIter.hasNext(); ) {
    var xmlElement = xmlElementIter.next();
    var tableName = xmlElement.getAttributeValue('table');
    for (var xmlRecordIter = xmlElement.getIterator('record');
      xmlRecordIter.hasNext(); ) {
      var xmlRecord = xmlRecordIter.next();
      var xmlFieldIter = xmlRecord.getIterator('field');
      for (var xmlField = xmlFieldIter.next();
        xmlFieldIter.hasNext(); ) {
        var fieldName = xmlField.getAttributeValue('name');
        for (var childIter = xmlField.getIteratorOverAllChildren();
          childIter.hasNext(); ) {
          var childElement = childIter.next();
          childElement.valueFor(tableName, fieldName);
        }
      }
    }
  }
}
```

XMLElementIterator - next()

Returns the next element in the iterator.

This method is available only in an Edge Encryption rule action script.

### Table 666: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 667: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLElement</td>
<td>The next XML element</td>
</tr>
</tbody>
</table>

This example is from the List Edit encryption rule.

```javascript
function ListEditAction(request) {
  var xmlContent = request.postParams.sysparm_xml.getAsXmlContent();
  for (var xmlElementIter = xmlContent.getIterator('record_update');
      xmlElementIter.hasNext(); ) {
    var xmlElement = xmlElementIter.next();
    var tableName = xmlElement.getAttributeValue('table');
    for (var xmlRecordIter = xmlElement.getIterator('record');
      xmlRecordIter.hasNext(); ) {
      var xmlRecord = xmlRecordIter.next();
    }
  }
}
```
var xmlRecord = xmlRecordIter.next();
for (var xmlFieldIter = xmlRecord.getIterator('field');
xmlFieldIter.hasNext(); ) {
    var xmlField = xmlFieldIter.next();
    var fieldName = xmlField.getAttributeValue('name');
    for (var childIter = xmlField.getIteratorOverAllChildren();
childIter.hasNext(); ) {
        var childElement = childIter.next();
        childElement.valueFor(tableName, fieldName);
    }
}

XMLElement API
The XMLElement class is available in Edge Encryption rule action scripts.

You get an XMLElement object by calling the next() method of an XMLElementIterator object.

XMLElement - getIterator(String xPath)
Returns an XMLElementIterator object for the XML element based on the specified parameter.
This method is available only in an Edge Encryption rule action script.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xPath</td>
<td>String</td>
<td>An XPath like expression that specifies where in the XMLElement object to start.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLElementIterator</td>
<td>An object that will iterate over elements in the XMLElement object.</td>
</tr>
</tbody>
</table>

This example is from the List Edit encryption rule.

function ListEditAction(request) {
  var xmlContent =
    request.postParams.sysparm_xml.getAsXmlContent();
  for (var xmlElementIter =
    xmlContent.getIterator('record_update');
xmlElementIter.hasNext(); ) {
    var xmlElement = xmlElementIter.next();
    var tableName = xmlElement.getAttributeValue('table');
    for (var xmlRecordIter = xmlElement.getIterator('record');
xmlRecordIter.hasNext(); ) {
        var xmlRecord = xmlRecordIter.next();
        for (var xmlFieldIter = xmlRecord.getIterator('field');
xmlFieldIter.hasNext(); ) {
            var xmlField = xmlFieldIter.next();
            var fieldName = xmlField.getAttributeValue('name');
            } } }

© 2017 ServiceNow. All rights reserved. 2691
XMLElement - getIteratorOverAllChildren()

Returns an XMLElementIterator object that includes all sub elements for the XML element based on the specified parameter.

This method is available only in an Edge Encryption rule action script.

Table 670: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 671: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLElementIterator</td>
<td>An object that will iterate over elements in the XMLElement object.</td>
</tr>
</tbody>
</table>

This example is from the List Edit encryption rule.

```javascript
function ListEditAction(request) {
    var xmlContent = request.postParams.sysparm_xml.getAsXmlContent();
    for (var xmlElementIter = xmlContent.getIterator('record_update');
        xmlElementIter.hasNext(); ) {
        var xmlElement = xmlElementIter.next();
        var tableName = xmlElement.getAttributeValue('table');
        for (var xmlRecordIter = xmlElement.getIterator('record');
            xmlRecordIter.hasNext(); ) {
            var xmlRecord = xmlRecordIter.next();
            for (var xmlFieldIter = xmlRecord.getIterator('field');
                xmlFieldIter.hasNext(); ) {
                var xmlField = xmlFieldIter.next();
                var fieldName = xmlField.getAttributeValue('name');
                for (var childIter = xmlField.getIteratorOverAllChildren();
                    childIter.hasNext(); ) {
                    var childElement = childIter.next();
                    childElement.valueFor(tableName, fieldName);
                }
            }
        }
    }
}
```
XMLElement - valueFor(String tableName, String fieldName)
Specifies that the value of the element maps to the specified field in the specified table.

Calling this method on an element value tells the proxy that the value for this element maps to the specified field in the specified table. The proxy then decides if the field needs to be encrypted.

This method is available only in an Edge Encryption rule action script.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>The table name</td>
</tr>
<tr>
<td>fieldName</td>
<td>String</td>
<td>The field name</td>
</tr>
</tbody>
</table>

This example is from the List Edit encryption rule.

```javascript
function ListEditAction(request) {
  var xmlContent = request.postParams.sysparm_xml.getAsXmlContent();
  for (var xmlElementIter = xmlContent.getIterator('record_update');
    xmlElementIter.hasNext(); ) {
    var xmlElement = xmlElementIter.next();
    var tableName = xmlElement.getAttributeValue('table');
    for (var xmlRecordIter = xmlElement.getIterator('record');
      xmlRecordIter.hasNext(); ) {
      var xmlRecord = xmlRecordIter.next();
      var fieldName = xmlRecord.getAttributeValue('field');
      for (var xmlFieldIter = xmlRecord.getIterator('field');
        xmlFieldIter.hasNext(); ) {
        var xmlField = xmlFieldIter.next();
        var fieldName = xmlField.getAttributeValue('name');
        for (var childIter = xmlField.getIteratorOverAllChildren();
          childIter.hasNext(); ) {
          var childElement = childIter.next();
          childElement.valueFor(tableName, fieldName);
        }
      }
    }
  }
}
```

XMLElement - encodedQueryFor(String tableName)
Specifies that the value of the element is an encoded query for the specified table.

Calling this function on an element tells the proxy that the value of the element is an encoded query for the specified table. The proxy parses the encoded query and encrypts the fields in the encoded query that need to be encrypted.

This method is available only in an Edge Encryption rule action script.
Table 674: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>A table name</td>
</tr>
</tbody>
</table>

Table 675: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

XMLElement - getName()
Returns the element name.
This method is available only in an Edge Encryption rule action script.

Table 676: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 677: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The element name</td>
</tr>
</tbody>
</table>

XMLElement - getAttributeValue(String attribute)
Returns the value of the specified attribute.
This method is available only in an Edge Encryption rule action script.

Table 678: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute</td>
<td>String</td>
<td>Attribute name</td>
</tr>
</tbody>
</table>
Global methods
Several global methods can be used in encryption rules.

print(String message)
Prints a message to the wrapper log file: <proxy server directory>/logs/wrapper_<date>.log.
This method is available only in an Edge Encryption rule action script.

Table 680: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>String</td>
<td>The message to be written to the wrapper log file.</td>
</tr>
</tbody>
</table>

Table 681: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

endsWith(String toMatch)
Brief description of the method.
This method is available only in an Edge Encryption rule action script.

Table 682: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toMatch</td>
<td>String</td>
<td>The string to match</td>
</tr>
</tbody>
</table>

Table 683: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>True if the specified string matches the end of the target string.</td>
</tr>
</tbody>
</table>

This example is from the Vacation encryption-rule condition.

```javascript
function VacationCondition(request) {
    if (endsWith(request.path, '/service_catalog.do') &
```
request.postParams.sysparm_action == 'execute_producer' &&
request.postParams.sysparm_id ==
'c322301653a33100c0eca5f4a11c084d')
return true;
return false;
}

string.replace(String old, String new)
Creates a new string with the specified existing sub-string replaced with the specified new sub-string.
This method is available only in an Edge Encryption rule action script.

Table 684: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>old</td>
<td>String</td>
<td>The sub-string to be replaced</td>
</tr>
<tr>
<td>new</td>
<td>String</td>
<td>The sub-string replace the old sub-string</td>
</tr>
</tbody>
</table>

Table 685: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The string with the specified sub-string replaced with the new sub-string.</td>
</tr>
</tbody>
</table>

Prohibited keywords
The Edge Encryption proxy validates encryption rule scripts before saving the rule. Many JavaScript keywords are not allowed in encryption rule scripts.

Table 686: Prohibited keywords

<table>
<thead>
<tr>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DIR</strong></td>
</tr>
<tr>
<td><strong>FILE</strong></td>
</tr>
<tr>
<td><strong>LINE</strong></td>
</tr>
<tr>
<td><strong>parent</strong></td>
</tr>
<tr>
<td><strong>proto</strong></td>
</tr>
<tr>
<td>Error</td>
</tr>
<tr>
<td>eval</td>
</tr>
<tr>
<td>getClass</td>
</tr>
<tr>
<td>getPrototypeOf</td>
</tr>
</tbody>
</table>
Dictionary attributes

You can add Edge Encryption dictionary attributes to tables and fields.

To set an Edge Encryption dictionary attribute to true, you must enter `attribute=true` in the Attributes field. To add a dictionary attribute to a record, see *Dictionary attributes* on page 1453.

Table 687: Edge Encryption dictionary attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Target element</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge Encryption Excluded</td>
<td>true/false</td>
<td>field or table</td>
<td>When set to true, the field or table cannot be encrypted.</td>
<td>False</td>
</tr>
<tr>
<td>[edge_encryption_excluded]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Target element</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Edge Encryption Enabled</td>
<td>true/false</td>
<td>field</td>
<td>When set to true, the field is eligible for encryption through an encryption configuration. Because this attribute is used by the system and cannot be modified, it is not always displayed to the user.</td>
<td>True for String fields</td>
</tr>
<tr>
<td>Edge Encryption Clear Text Allowed</td>
<td>true/false</td>
<td>field</td>
<td>When set to true, allows server-side scripts to append non-encrypted data to an encrypted string within the field.</td>
<td>False</td>
</tr>
</tbody>
</table>

**Note:**
This attribute does not indicate that a field is encrypted, nor does it trigger any encryption logic on the field. Rather, the attribute determines the possibility of the field being encrypted by a user.

**Installed with Edge Encryption**

Several types of components are installed with the Edge Encryption feature.
Tables installed with Edge Encryption

**Tables**

Edge Encryption adds the following tables.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge Encryption Configuration</td>
<td>Contains encrypted fields and tables for which attachments are encrypted.</td>
</tr>
<tr>
<td>[sys_encryption_configuration]</td>
<td></td>
</tr>
<tr>
<td>Edge Encryption Rule</td>
<td>Contains a record for each rule. A rule has a name, the condition when it is used, a script, and an order field.</td>
</tr>
<tr>
<td>[sys_encryption_rule]</td>
<td></td>
</tr>
<tr>
<td>Edge Encryption Invalid Insert Log</td>
<td>Contains log messages created for attempts to save unencrypted data to an encrypted field.</td>
</tr>
<tr>
<td>[sys_edge_encryption_invalid_insert_log]</td>
<td></td>
</tr>
<tr>
<td>Edge Encryption Proxy</td>
<td>Contains information about the encryption proxy application.</td>
</tr>
<tr>
<td>[sys_encryption_proxy]</td>
<td></td>
</tr>
<tr>
<td>Edge Proxy Encryption Type</td>
<td>Used for enabling and disabling encryption types on the encryption form.</td>
</tr>
<tr>
<td>[sys_proxy_encryption_type]</td>
<td></td>
</tr>
<tr>
<td>Encryption Job Execution</td>
<td>Supports mass encryption jobs.</td>
</tr>
<tr>
<td>[sys_encryption_job_execution]</td>
<td></td>
</tr>
<tr>
<td>Encryption Job Execution Chunk</td>
<td>Supports mass encryption jobs.</td>
</tr>
<tr>
<td>[sys_encryption_job_execution_chunk]</td>
<td></td>
</tr>
<tr>
<td>Scheduled Encryption Job</td>
<td>Lists scheduled jobs for encryption, decryption, key rotation, order token repair, and database recovery.</td>
</tr>
<tr>
<td>[sysauto_encryption_job]</td>
<td></td>
</tr>
</tbody>
</table>

**Properties Installed with Edge Encryption**

**Properties**

Edge Encryption adds the following properties.

**Note:** To open the System Properties [sys_properties] table, enter `sys_properties.list` in the navigation filter.
Table 689: Properties for Edge Encryption

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sn_edge_encryption.logging.destination</td>
<td>Where log messages are to go.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: string</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: file</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: System Properties [sys_properties] table</td>
</tr>
<tr>
<td>sn_edge_encryption.logging.verbosity</td>
<td>The logging level to use.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type</strong>: string</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value</strong>: info</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location</strong>: System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>

Roles installed with Edge Encryption

Table 690: Roles installed with Edge Encryption

<table>
<thead>
<tr>
<th>Role title</th>
<th>Description</th>
<th>Contains roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>edge_encryption</td>
<td>Edge Encryption proxies log into the instance as a user, with a user name and password. The user must have this role assigned.</td>
<td>None</td>
</tr>
</tbody>
</table>

Notifications

Notifications keep users informed of events that concern them. The system can notify users by email, SMS text message, or push notification. It can also integrate with telephony providers to manage phone calls and SMS messages.

What’s new

- Email and notifications release notes

Email

- Basic email setup on page 2702
- Advanced email setup on page 2705
- Create an email notification on page 2756
- The email client on page 2857
- System mailboxes
- Inbound email actions on page 2822

Push Notifications

- Set up push notifications for the ServiceNow mobile app on page 2876
- Set up push notifications for a custom push app on page 2880
- Push notifications on page 2869

Subscription-based notifications

Notify

Troubleshoot and get help
Email notifications

Use email notifications to send selected users email about specific activities in the system, such as updates to incidents or change requests.

If you want to change how the instance processes incoming email, see Inbound email actions on page 2822.

Email notifications allow administrators to specify:

• When to send the notification
• Who receives the notification
• What content is in the notification

Additional email notification options are available. Users can subscribe to notifications, and administrators can make some notifications mandatory.

Administrators also have the option of converting existing email notifications to a rich HTML format. This format provides several advantages, including:

• Raw HTML content is converted into a WYSIWYG format.
• The content can be edited in a feature-rich HTML editor.
• Mail scripts are condensed into a single, easy-to-read line that can be reused in multiple email notifications.
• To prevent broken links, items like images and incidents, that are linked with URLs relative to a particular instance are converted to absolute links. For example, if an incident is linked using a relative URL, the link is converted to an absolute link.

Note: The rich HTML format is the default for all new email notifications.

See the Email logs on page 1059 for examples of what you might see if a notification or inbound email action is not processed.

Note: Instances cannot send or receive encrypted email messages. The system strips out the body of the encrypted email because it cannot process the encrypted content in plain text or HTML.

Email setup

All email notifications use the email properties that you define and the email accounts that you set up.

Email accounts

Email administrators set up email accounts to allow the system to connect to external mail services such as POP3, SMTP, or IMAP servers. You can use the ServiceNow-provided email accounts or create your own accounts for your own email services.
Email properties

Use *Email Properties* to configure settings for inbound and outbound email. Email properties apply to all email accounts.

Email size restrictions

For instances that use a *standard email configuration*, the system cannot send or receive emails that are larger than 25MB, including the email header, body text, and attachments. Instances configured to use an *alternate email configuration* may support different maximum total file sizes. The maximum email size limit is enforced regardless of any configured attachment size limits.

**Note:** Even if the email and its attachments are less than 25MB, the actual size of an email message could be larger when it gets encoded by an email client. So these messages might exceed the limit when they reach the instance.

Email service availability

Email service is intentionally shut down on instance clones to prevent resending already delivered email. Upgrades no longer require an extended interruption of email service. Instances continue to process email during an upgrade. For a possible explanation for undelivered emails, see the blog post *Whitelisting emails sent from the ServiceNow infrastructure* by a ServiceNow Technical Support Engineer in the ServiceNow Community.

For instructions on creating and sending custom emails when events on the instance occur, see the tasks in *Email notifications* on page 2701.

Instance-to-instance communication via email

ServiceNow does not recommend using email notifications to communicate between two instances. Use *Web services* on page 3044 instead.

Basic email setup

All production instances can send and receive email using ServiceNow-provided resources.

Basic email offers the following services and features:

- Mail servers maintained by ServiceNow.
  - Encrypt mail with opportunistic TLS (Transport Layer Security) if supported by your mail servers.
  - Provide a dedicated mailbox for your instance.
- Pre-configured email accounts to connect to ServiceNow mail servers.
  - An SMTP account sends email to your primary Mail Exchange (MX) server from your instance email address of instance@service-now.com.
  - A POP3 account receives email sent to your instance email address of instance@service-now.com.
- High availability features from ServiceNow datacenters.
- Spam detection for incoming email.

Administrators who want to use basic email services can do so by enabling the email properties for sending and receiving email.
Enable basic email
Enable basic email to use ServiceNow-provided email servers and accounts.

Role required: admin

1. Navigate to System Properties > Email.
2. Configure these email properties and click Save.

### Table 691: Email Property Descriptions

<table>
<thead>
<tr>
<th>Property section</th>
<th>Label</th>
<th>System property</th>
<th>Setting required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound Email Configuration</td>
<td>Email sending enabled</td>
<td>glide.email.smtp.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Inbound Email Configuration</td>
<td>Email receiving enabled</td>
<td>glide.email.read.active</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Network layout basic email
Basic email uses ServiceNow-provided mail servers and email accounts. The instance has an email address of instance@service-now.com.
Figure 755: Standard email configuration
Advanced email setup

With an advanced email setup, you can use your own SMTP server, POP3 server, or both.

Setting up your own email environment can be useful if you want to use existing filtering, retention, or compliance aspects of your internal email architecture. You can set up email in several ways:

- Use your own SMTP server to forward email to ServiceNow servers.
- Use your own SMTP server to send email.
- Use your own POP3 server to receive email.
- Use your own SMTP and POP3 servers to send and receive email.
- Use an OAuth 2.0-enabled SMTP server to send email from a third-party service.
- Use an OAuth 2.0-enabled IMAP server to receive email from a third-party service.

Enable email forwarding to instance

You can use your own SMTP server to forward email to the ServiceNow-provided email address.

- Role required: admin
- Email server required: SMTP

1. Configure your SMTP server to forward email sent to a custom address to the actual instance email address. For example, forward mail sent to service-desk@company.com to instance@service-now.com.

2. Navigate to System Properties > Email.

3. Configure these email properties and click Save.

<table>
<thead>
<tr>
<th>Property section</th>
<th>Label</th>
<th>System property</th>
<th>Setting required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound Email Configuration</td>
<td>Email sending enabled</td>
<td>glide.email.smtp.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Inbound Email Configuration</td>
<td>Email receiving enabled</td>
<td>glide.email.read.active</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Enable using your own SMTP server

You can use your own SMTP server to send email from the instance.

- Role required: admin
- Email server required: SMTP

1. Optional: To receive email sent to a custom address, configure your SMTP server to forward email sent to the custom address to the actual instance email address. For example, forward mail sent to service-desk@company.com to instance@service-now.com.

2. Navigate to System Mailboxes > Administration > Email Accounts. The system displays the list of available email accounts.

3. Locate the record for ServiceNow SMTP and change Active to false.
4. Click **New**. The system displays a blank Email Account form.

5. Create an email account record for your SMTP server where the **Type** is **SMTP**.

6. From **Related Links**, click **Test SMTP connection**. If the email account is valid, the system returns a **Completion code** of **Success**.

7. Navigate to **System Mailboxes > Administration > Email Properties**. The system displays the list of email properties.

8. Configure this email property.
Table 693: Outbound email property

<table>
<thead>
<tr>
<th>Property Section</th>
<th>Label</th>
<th>System Property</th>
<th>Setting Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound Email</td>
<td>Email sending enabled.</td>
<td>glide.email.smtp.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9. Optional: If you want the instance to receive email, configure this email property.

Table 694: Inbound email property

<table>
<thead>
<tr>
<th>Property Section</th>
<th>Label</th>
<th>System Property</th>
<th>Setting Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound Email</td>
<td>Email receiving enabled</td>
<td>glide.email.read.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. Click **Save**.

**Enable using your own POP3 server**
You can use your own POP3 server to store and receive email for the instance.

- Role required: admin
- Email server required: POP3

1. On your POP3 server, create a mailbox for your instance that has a custom email address. For example, create a mailbox for **service-desk@company.com**.
2. Navigate to **System Mailboxes > Administration > Email Accounts**. The system displays the list of available email accounts.
3. Optional: If you do not want to receive email sent to the instance@service-now.com mailbox, locate the record for **ServiceNow POP3** and change **Active** to **false**.
   
   An instance can receive email from multiple POP3 accounts at the same time. Leaving the **ServiceNow POP3** account active allows the instance to receive email sent to the instance default email address.
4. Click **New**.
   The system displays a blank Email Account form.

5. Create an email account record for your POP3 server where the **Type** is **POP3**.

6. From **Related Links**, click **Test POP3 connection**.
   If the email account is valid, the system returns a **Completion code** of **Success**.

7. Navigate to **System Mailboxes > Administration > Email Properties**.
   The system displays the list of email properties.

8. Configure this email property.
Table 695: Inbound email property

<table>
<thead>
<tr>
<th>Property Section</th>
<th>Label</th>
<th>System Property</th>
<th>Setting Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound Email</td>
<td>Email receiving</td>
<td>glide.email.read.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration</td>
<td>enabled</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9. Optional: If you want to send email, configure this email property.

Table 696: Outbound email property

<table>
<thead>
<tr>
<th>Property Section</th>
<th>Label</th>
<th>System Property</th>
<th>Setting Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound Email</td>
<td>Email sending</td>
<td>glide.email.smtp.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration</td>
<td>enabled.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. Click **Save**.

*Enable using your own SMTP and POP3 servers*

You can use your own SMTP and POP3 servers to send email from the instance and to store and receive email for the instance.

- Role required: admin
- Email servers required:
  - SMTP
  - POP3

1. On your POP3 server, create a mailbox for your instance. For example, create a mailbox for service-desk@company.com.

2. Navigate to **System Mailboxes > Administration > Email Accounts**.
   The system displays the list of available email accounts.

3. Locate the record for **ServiceNow SMTP** and change **Active** to **false**.
4. Optional: If you do not want to receive email sent to the instance@service-now.com mailbox, locate the record for ServiceNow POP3 and change **Active** to **false**.

An instance can receive email from multiple POP3 accounts at the same time. Leaving the ServiceNow POP3 account active means that the instance receives email sent to its default email address.

<table>
<thead>
<tr>
<th>Email Accounts</th>
<th>New</th>
<th>Go to</th>
<th>Name</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ServiceNow POP3</strong></td>
<td></td>
<td></td>
<td><strong>true</strong></td>
<td><strong>POP3</strong></td>
</tr>
<tr>
<td><strong>ServiceNow SMTP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Click **New**.

The system displays a blank Email Account form.

6. Create an email account record for your SMTP server where the **Type** is **SMTP**.

7. From **Related Links**, click **Test SMTP connection**.

If the email account is valid, the system returns a **Completion code** of **Success**.
8. Click **New**.
   The system displays a blank Email Account form.

9. Create an email account record for your POP3 server where the **Type** is **POP3**.

10. From **Related Links**, click **Test POP3 connection**.
    If the email account is valid, the system returns a **Completion code** of **Success**.

11. Navigate to **System Mailboxes** > **Administration** > **Email Properties**.
    The system displays the list of email properties.

12. Configure these email properties.
Table 697: Email properties

<table>
<thead>
<tr>
<th>Property Section</th>
<th>Label</th>
<th>System Property</th>
<th>Setting Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound Email</td>
<td>Email sending enabled</td>
<td>glide.email.smtp.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inbound Email</td>
<td>Email receiving enabled</td>
<td>glide.email.read.active</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13. Click **Save**.

**OAuth email authentication**

OAuth enables your instance to receive and send email through a third-party email account.

Open Authentication (OAuth) is an open standard for authorization that provides administrators with an authorization method when connecting to incoming IMAP and outgoing SMTP servers. OAuth enables an instance to receive and send email from a third-party account, such as Gmail, without having to enter the credentials for that account.

The OAuth 2.0 implementation requires you to obtain an access and refresh token from your third-party email provider for each third-party email account. The tokens are automatically saved to the instance database. They provide authorization for all email communication between the instance and the authorized third-party account. A scheduled job regularly checks to see if email access tokens are valid. If the access token is not valid, but the refresh token is, the instance automatically regenerates a new access token.

OAuth 2.0 support is available starting with the Geneva release. The OAuth implementation supports IMAP and SMTP accounts only. POP3 is not supported. OAuth 1.0 is no longer supported.

---

**Note:** Customer email accounts configured to use OAuth 1.0 authentication with Gmail cease to function as of April 20, 2015. See [KB0546976](#) for more information.

Activating the Email - OAuth support for IMAP and SMTP plugin allows you to use OAuth with email. If you upgrade to Geneva or later instances and are already using OAuth 1.0, activate the plugin again.

See **Outbound REST with OAuth 2.0 profile tutorial - integrating with Google Contacts API** on page 3408 for an example of using an OAuth 2.0 profile to authenticate an outbound REST message with Google to retrieve contact information. Also see **OAuth 2.0** on page 1952 for more information on OAuth 2.0 support in the instance.

Activate the OAuth email authentication plugin

OAuth email authentication requires the Email - OAuth support for IMAP and SMTP plugin.

Role required: admin

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.

3. Optional: If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click **Activate**.

Enable OAuth 2.0 for email

Setting up OAuth 2.0 for email requires you to obtain access and refresh tokens from your email provider.
• Role required: admin
• Plugin required: Email - OAUTH support for IMAP and SMTP

1. Log in to your third-party email account, such as Gmail, and enable OAuth 2.0.
2. Obtain the following from your third-party email account:
   • client ID
   • client secret
   • authorization URL
   • token URL
   • redirect URL
   • token revocation URL

3. Navigate to System OAuth > Application Registry.
4. Click New.
5. Click Connect to a third party OAuth Provider to create an application registry record that email uses.
6. Use the information you obtained from your third-party email account to fill in the fields on the form.
   See Use a third-party OAuth provider on page 1958 for instructions. Create the OAuth application registry record and its associated OAuth Entity Profile and OAuth Entity Scope records.
7. Click Submit.
8. Navigate to System Mailboxes > Administration > Email Accounts.
   The system displays the list of available email accounts.
9. Optional: If you do not want to receive email sent to the default instance email address, locate the record for ServiceNow POP3 and change Active to false.
   The system can receive email from multiple POP3 email accounts.
10. Locate the records for ServiceNow SMTP and change Active to false.
11. Click New.
   The system displays a blank Email Account form.
12. Create an email account record for your OAuth 2.0 SMTP server where the Type is SMTP.
13. For Authentication, select OAuth 2.0.
14. For OAuth Profile, select the application registry record you created.

15. Click Authorize Email Account Access to obtain the access and refresh tokens.

   Another browser window opens asking you to authorize the account access on the third-party email account.

16. Authorize the access.

   After the authorization is successful and the tokens are saved to the instance, the Authorize Email Account Access button no longer appears on the Email Account form.

17. Click New.

   The system displays a blank Email Account form.

18. Create an email account record for your OAuth 2.0 IMAP server where the Type is IMAP.

   Use the same Authentication and OAuth profile settings as the OAuth SMTP email account.

---

**Network layout email forwarding**

You can use your own SMTP server to forward email sent to a custom email address to the actual instance email address.

Typically, organizations use a custom email address to change the reply-to address from the instance name to a custom email address within their domain. This configuration requires an SMTP server in your internal email architecture to forward email sent to your custom email address to the actual instance email address. Implement a spam filter on your custom email address to prevent unwanted mail from triggering actions on your instance.
Network layout using your own SMTP server
You can use your own SMTP server to send email to and from your instance.

This layout bypasses the ServiceNow-provided SMTP server normally used to send email from your instance.
Figure 757: Sending email using your own SMTP server

ServiceNow Instance

Outbox with Email To: user@yourdomain.com

Send

Receive

ServiceNow Incoming Mail Server
pop.service-now.com
Route To Mailbox For: instance@service-now.com

Domain Name System (DNS)

Lookup Mail Server For: yourdomain.com
mx.yourdomain.com

Domain Name System (DNS)

Lookup Mail Server For: service-now.com
mx.service-now.com

Your Outbound Mail Server
smtp.yourdomain.com
Sends Email To: user@yourdomain.com

Your Incoming Mail Server
pop.yourdomain.com
Route To Mailbox For: user@yourdomain.com

Yes

Email Passes Your Spam Filter?

No

Stop

User Email Account
Inbox For: user@yourdomain.com
Outbox with Email To: helpdesk@yourdomain.com

Send
Network layout using your own POP3 server
You can use your own POP3 server to store and retrieve email for your instance.

This layout bypasses the ServiceNow-provided POP3 server normally used to store and retrieve email for your instance.
Network layout using your own SMTP and POP3 servers
You can use your own SMTP and POP3 server to send email from and store and retrieve email for your instance.

This layout bypasses both the ServiceNow-provided SMTP and POP3 servers normally used to send, store, and retrieve email for your instance.
Geneva ServiceNow ServiceNow Platform

Figure 759: Using your own SMTP and POP3 servers

ServiceNow Instance

Outbox with Email To:
user@yourdomain.com

Send

Domain Name System (DNS)

Lookup Mail Server For:
yourdomain.com
mx.yourdomain.com

Receive

Domain Name System (DNS)

Lookup Mail Server For:
yourdomain.com
mx.yourdomain.com

Your Outbound Mail Server
smtp.yourdomain.com
Send Email To:
user@yourdomain.com

Your Incoming Mail Server
pop.yourdomain.com
Route To Mailbox For:
user@yourdomain.com

Your Incoming Mail Server
pop.yourdomain.com
Route To Mailbox For:
helpdesk@yourdomain.com

Your Outbound Mail Server
smtp.yourdomain.com
Send Email To:
helpdesk@yourdomain.com

Stop

Email Passes Your Spam Filter?

No

Outbox with Email To:
helpdesk@yourdomain.com

Receive

User Email Account

Inbox For:
user@yourdomain.com

Send

© 2017 ServiceNow. All rights reserved. 2721
Email accounts

Email accounts store the connection details and credentials the system uses to access external email servers and services.

By default, instances come with email accounts for ServiceNow-managed SMTP and POP3 servers to send and receive email. These accounts cannot be modified, but can be deactivated if you do not want to use them. To connect to other email servers and services you must create email accounts for them. You can create email accounts for servers and services that use these protocols:

- SMTP
- POP3
- IMAP

**Note:** The system only allows one SMTP email account to be active at a time and sends all email through this account. You can however receive email from multiple POP3 or IMAP accounts.

The system stores individual email accounts in the `sys_email_account` table. Create separate email accounts to send and receive email. Use `email properties` to define how the system processes email for all email accounts.

By default, the ServiceNow POP3 server provides each instance with its own mailbox at the address instance@service-now.com. Likewise, the ServiceNow SMTP server sends email from the address instance@service-now.com. To create another mailbox, provision your own POP3 or IMAP server and create an email account to the server on the instance. To change the sent from email address, update the address in the email account you use for sending email.
Figure 760: A ServiceNow configured email account

Configure an email account
You can create email accounts in addition to the accounts provisioned for you.

- Role required: admin
- Email server: a compatible email server
  - SMTP
  - POP3
  - IMAP

1. Navigate to System Mailboxes > Administration > Email Accounts.
2. Click New.
3. Fill in the fields on the form (see table).
4. Click **Submit**.
5. Click the test connection link at the bottom to test the SMTP, POP3, or IMAP account.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name to identify this Email Account.</td>
</tr>
<tr>
<td>Type</td>
<td>Mail server type. The choices are:</td>
</tr>
<tr>
<td></td>
<td>• Email Reader</td>
</tr>
<tr>
<td></td>
<td>• POP3</td>
</tr>
<tr>
<td></td>
<td>• IMAP</td>
</tr>
<tr>
<td></td>
<td>• Email Sender</td>
</tr>
<tr>
<td></td>
<td>• SMTP (only one active account permitted)</td>
</tr>
<tr>
<td>Authentication</td>
<td>The type of authentication used for the email account to connect to the email server. The choices are <strong>Password</strong>, <strong>OAUTH</strong>, and <strong>OAUTH 2.0</strong>. The Email - OAuth support for IMAP and SMTP plugin must be active for the OAuth options to be visible.</td>
</tr>
<tr>
<td>OAuth Provider</td>
<td>Select the OAuth application registry record for this account. This field appears if you selected <strong>OAuth 2.0</strong>.</td>
</tr>
<tr>
<td>Server</td>
<td>Remote Server to which this account connects.</td>
</tr>
<tr>
<td>Active</td>
<td>Determines if this Email Account is active.</td>
</tr>
<tr>
<td>ServiceNow Configured</td>
<td>Indicates if this account is provisioned by ServiceNow. This field is read-only. If you create an account, this option is not selected.</td>
</tr>
<tr>
<td>Email user label</td>
<td>A display value used for outgoing messages. This field is for SMTP type accounts only.</td>
</tr>
<tr>
<td>User name</td>
<td>The user name or ID to authenticate an email address. The value in this field is also the <strong>From</strong> address when the instance sends email. If you are using SMTP, this <strong>must</strong> be a full email address. <strong>Note:</strong> The address in the <strong>From</strong> field on the Notification form takes precedent over this field.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Password</td>
<td>Password when Authentication type is Password.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You may need to increase the size of this field to accommodate longer passwords. By default, this field has a size of 40.</td>
</tr>
<tr>
<td>Enable SSL</td>
<td>Enables Secure Socket Layers when connecting to an Email Server.</td>
</tr>
<tr>
<td>Enable TLS</td>
<td>Enables Transport Layer Security when connecting to an Email Server.</td>
</tr>
<tr>
<td>Port</td>
<td>Connection TCP port.</td>
</tr>
</tbody>
</table>

![Email Account](image-url)
Email properties

The Email Properties page is where you can configure settings for inbound and outbound email.

Email properties are available from either of these modules:

- **System Mailboxes > Email Properties**
- **System Properties > Email Properties**
<table>
<thead>
<tr>
<th><strong>Outbound Email Configuration</strong></th>
<th><strong>Inbound Email Configuration</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Email sending enabled</td>
<td>Email receiving enabled</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Send all email to this test email address (non-production testing)</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Append timezone to dates and times in sent email</td>
<td>Identify email as a reply to these subject prefixes</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Create visible watermark in sent email. If false, create invisible watermark via hidden div tag</td>
<td>Identify email as a forward by these subject prefixes</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Resend email if server returns these SMTP error codes 421, 450, 451, 452</td>
<td>Discard everything below this text if found in a reply body (comma separated, case sensitive)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Do not resend email if server returns these SMTP error codes 500, 501, 502, 503, 504, 505, 506, 502, 503, 504</td>
<td>Ignore email with these headers (comma separated, case insensitive)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Resend email when server returns unknown SMTP error codes</td>
<td>Ignore email when subject starts with text (comma separated, case insensitive)</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Notes that can view email in the Activity formatter when including “Sent/Received Emails”</td>
<td>Ignore email from these senders. Use the name before the @ sign. (comma-separated)</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Number of journal entries (Additional comments, Work notes, etc.) included in email notifications (-1 means all)</td>
<td>Automatically create users for incoming emails from trusted domains</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>Default password for users created from email sent from trusted domains. (must reset upon login)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>password</td>
</tr>
<tr>
<td>Trusted domains when creating new users from incoming email (ignore email from untrusted domains unless from an existing user. use * for all domains)</td>
<td></td>
</tr>
</tbody>
</table>
Email accounts are configured in the System Mailboxes > Administration > Email Accounts module, starting with the Geneva release. See Configure an email account on page 2723 for instructions.

Email diagnostics are available from the System Mailboxes > Email Diagnostics module, starting with the Geneva release.

Outbound mail configuration
The Outbound Mail Configuration section of the Email Properties page contains properties for sending email.
## Outbound Email Configuration

**Email sending enabled**
- Yes | No

**Send all email to this test email address (non-production testing)**

**Append timezone to dates and times in sent email**
- Yes | No

**Create visible watermark in sent email. If false, create invisible watermark via hidden div tag.**
- Yes | No

**Resend email if server returns these SMTP error codes**
- 421,450,451,452

**Do not resend email if server returns these SMTP error codes**
- 500,501,502,503,504,550,551,552,553,554

**Resend email when server returns unknown SMTP error codes.**
- Yes | No

**Roles that can view email in the Activity formatter when including "Sent/Received Emails"**
- itil

**Number of journal entries (Additional comments, Work notes, etc.) Included in email notifications (-1 means all).**
- 3
<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.smtp.active</td>
<td>Email sending enabled</td>
<td>Specifies whether to enable or disable the outgoing mail server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td>glide.email.test.user</td>
<td>Send all email to this test email address</td>
<td>Specifies the comma-separated list of email addresses to which the instance sends all email messages. Typically used in non-production instances for testing purposes.</td>
</tr>
<tr>
<td></td>
<td>(non-production testing)</td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td>glide.email.append.timezone</td>
<td>Append time zone to dates and times in sent mail</td>
<td>Specifies whether to append the system time zone to date and date/time values in outbound emails. For example, 2010-07-02 04:01:14 PST.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td>glide.email.watermark.visible</td>
<td>Create visible watermark in sent mail</td>
<td>Indicates whether the watermark in email notifications is visible (true) or wrapped in a hidden div tag (false).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td>glide.smtp.defer_retry_ids</td>
<td>Resend email if server returns these SMTP error codes</td>
<td>Specifies the comma-separated list of SMTP error codes that force the instance to resend email.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: 421,450,451,452</td>
</tr>
<tr>
<td>glide.smtp.fail_message_ids</td>
<td>Do not resend email if server returns these SMTP error codes</td>
<td>Specifies the comma-separated list of SMTP error codes that prevent the instance from resenting email.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: 500,501,502,503,504,505,551,552,553,554</td>
</tr>
<tr>
<td>Property</td>
<td>Label</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| glide.smtp.default_retry      | Resend email when server returns unknown SMTP error codes.            | Enables (true) or disables (false) resending email when an unknown SMTP error code is encountered. The instance only recognizes the SMTP error codes defined in the glide.smtp.defer_retry_ids property.  
  • Type: true | false  
  • Default value: true |
| glide.ui.activity.email_roles | Roles that can view email in the Activity formatter when including "Sent/Received Emails" | Specifies the comma-separated list of roles that can view email in the activity formatter.  
  • Type: string  
  • Default value: itil |
| glide.email.journal.lines     | Number of journal entries (Additional comments, Work notes, etc.) included in email notifications (-1 means all). | Specifies the number of entries from a journal field, such as Additional comments or Work notes, included in email notifications. A value of -1 includes all journal entries.  
  • Type: integer  
  • Default value: 3  
  • Learn More: Restrict the Number of Entries Sent in a Notification |

**Inbound mail configuration**

The **Inbound Mail Configuration** section of the Email Properties page contains properties to control inbound email.
## Inbound Email Configuration

**Email receiving enabled**

- Yes | No

**Identify email as a reply by these subject prefixes**

re:, aw:, r:, Accepted:, Tentative:, Declined:

**Identify email as a forward by these subject prefixes**

fw:, fwd:

**Discard everything below this text if found in a reply body (comma separated, case sensitive)**

```plaintext
\n\n---- Original Message ----, \n\n________ \n\nFrom:
```

**Ignore email with these headers (comma separated name:value pairs)**

X-ServiceNow-Spam-Flag:YES,X-ServiceNow-Virus:INFECTED,Auto-Resp:

**Ignore email when subject starts with text (comma separated, case insensitive)**

out of office autoreply, undeliverable:, delivery failure:, returned mail:, aut:

**Ignore email from these senders. Use the name before the @ sign. (comma-separated)**

mailer-daemon, postmaster

**Automatically create users for incoming emails from trusted domains**

- Yes | No

**Default password for users created from email sent from trusted domains. (must reset upon login)**

password
<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.read.active</td>
<td>Email receiving enabled</td>
<td>Specifies whether to enable or disable the inbound mail server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td>glide.email.reply_subject_prefix</td>
<td>Identify email as a reply by these subject prefixes</td>
<td>Specifies the comma-separated list of prefixes in the subject line that identify an email reply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Default value: re:, aw:, r:, Accepted:, Tentative, Declined</td>
</tr>
</tbody>
</table>

**Note:** The case of the reply prefix in the email, for example RE:, must exactly match the case of the prefixes defined in this property. If, for example, an email contains the Re: prefix and only RE: is defined in the property, the email will not be recognized as a reply. Therefore, it is a best practice to define multiple versions of the prefix, including mixed-case versions, such as RE:, Re:, and so on.
<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.email.forward_subject_prefix | Identify email as a forward by these subject prefixes | Specifies the comma-separated list of prefixes in the subject line that identify a forwarded email.  
- Type: string  
- Default value: fw:, fwd:  
  
**Note:** The case of the forward prefix in the email, for example fw:, must exactly match the case of the prefixes defined in this property. If, for example, an email contains the Fwd: prefix and only fwd: is defined in the property, the email will not be recognized as a forward. Therefore, it is a best practice to define multiple versions of the prefix, including mixed-case versions, such as FWD:, Fwd:, and so on. |
| glide.pop3.reply_separators | Discard everything below this text if found in a reply body (comma separated, case sensitive) | Specifies the comma-separated list of separators that cause the instance to disregard everything below the text string in the message body. This list is case sensitive.  
- Type: string  
- Default value: \n\n-----, \n_____  
Original Message-----, \n\n_____\n\nFrom: |
<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.pop3.ignore_headers | Ignore email with these headers (comma separated name:value pairs) | Specifies the comma-separated list of email headers that cause the instance to ignore an email message. Use the format name:value to specify email header types and values. You can use a wildcard (*) for the subtype. For example, Content-Type:multipart/*; report-type=delivery-status; ignores emails containing a type of multipart and a value of report-type=delivery-status. For syntax specifications, see http://www.w3.org/Protocols/rfc1341/4_Content-Type.html.  
- Type: string  

**Note:** If the Email Filters plugin is activated, the **Ignore header** email filter overrides this property. The property does not appear on the Mail Properties page when email filters are enabled.
<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.pop3.ignore_subjects | Ignore email when subject starts with text (comma separated, case insensitive) | Specifies the comma-separated list of strings that cause the instance to ignore an email message if the string is present at the start of a subject line. These values are not case sensitive.  
  - Type: string  
  - Default value: out of office autoreply, undeliverable:, delivery failure:, returned mail:, autoreply                                                                                   |
<p>|                        |                                                                      | <strong>Note:</strong> If the Email Filters plugin is activated, the <strong>Ignore subject</strong> email filter overrides this property. The property does not appear on the Mail Properties page when email filters are enabled.                                                                                     |
| glide.pop3.ignore_senders | Ignore email from these senders. Use the name before the @ sign. (comma-separated) | Specifies the comma-separated list of senders that cause the instance to ignore an email message. Enter only the name before the at sign (@).                                                                                                                                          |
|                        |                                                                      | <strong>Note:</strong> If the Email Filters plugin is activated, the <strong>Ignore sender</strong> email filter overrides this property. The property does not appear on the Mail Properties page when email filters are enabled.                                                                                     |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.pop3readerjob.create_caller</td>
<td>Automatically create users for incoming emails from trusted domains</td>
<td>Controls the behavior when an instance receives an email from an email address not associated with a user record. If this property is set to true, the instance creates a new user record for the email address and places that new user in the Caller field of any tickets created. If the property is set to false, the instance places Guest in the Caller field of any tickets created.</td>
</tr>
</tbody>
</table>
|                                 |                                                                        | • Type: true | false  
|                                 |                                                                        | • Default value: false  
|                                 |                                                                        | • Learn More: Enabling Automatic User Creation                                                                                                                                                    |
| glide.user.default_password      | Default password for users created from email sent from trusted domains. (must reset upon login) | Specifies the password for new users created from incoming email. Users must reset the password at first login.  
|                                 |                                                                        | • Type: string  
|                                 |                                                                        | • Default value: password  
|                                 |                                                                        | • Learn More: Enabling Automatic User Creation                                                                                                                                                    |
| glide.user.trusted_domain        | Trusted domains when creating users from incoming email (Ignore mail from untrusted domains unless from an existing user; use * for all domains) | Specifies the comma-separated list of trusted domains for creating users from incoming emails. Use an asterisk (*) to trust all domains. The instance ignores incoming email from other domains unless it is from an existing user’s address. The instance does not create guest users from email from untrusted domains. |
|                                 |                                                                        | • Type: string  
|                                 |                                                                        | • Default value: asterisk (*)  
|                                 |                                                                        | • Learn More: Enabling Automatic User Creation                                                                                                                                                    |

**Additional email properties**

There are several advanced email properties that you can use to fine tune the way your instance sends or receives email.

You must add these properties to the System Property [sys_properties] table before they can be used to overwrite the defaults.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| com.glide.email.max_body_bytes | Sets the maximum body size in bytes allowed per inbound email.  
  • Type: integer  
  • Default value: 1048576 |
| com.glide.email.max_read | Specifies the maximum number of emails a POP3 reader should process concurrently.  
  • Type: integer  
  • Default value: 20 |
| com.snc.on_call_rotation.reminders.showtz | Specifies whether to show a user's timezone.  
  • Type: true | false  
  • Default value: false |
| glide.email.inbound.calendar_behavior | Specifies how the system stores calendar data, such as an invitation or an invitation response. Enter one of these options (not case sensitive):  
  • Attach: Store the calendar data as an attachment on the associated record, such as the incident or change that triggers an invitation.  
  • Ignore: Discard the calendar data.  
  • Inline: Store the calendar data as text in the email Body field.  
  • Type: string  
  • Default value: Attach |
| glide.email.inbound.convert_html_inline_attachment_references | Specifies whether to convert inbound email HTML so email images appear in the email HTML body preview. The system displays broken cid (content ID) links in place of images received when this property is disabled. The format in which the system displays an email image depends on the property setting at the time the email is received, not the current property setting.  
  • Type: true | false  
  • Default value: true |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.email.inbound.max_attachment_count     | Sets the maximum number of attachments allowed per inbound email. This property is available starting with the Eureka Patch 4 release.  
• Type: integer  
• Default value: 30  
• Location: System Property [sys_properties] table |
| glide.email.inbound.max_total_attachment_size | Sets the maximum total attachment size in bytes allowed per inbound email. This property is available starting with the Eureka Patch 4 release.  
• Type: integer  
• Default value: 18874368  
• Location: System Property [sys_properties] table |
| glide.email.mail_to                          | Specifies the email address for sending notifications that use the ${mailto:} variable.  
• Type: string  
• Default value: value of glide.email.user property |
| glide.email.name_split                       | Specifies the delimiter used between first and last names in an email address. For example, a delimiter of "." (period) in the email address john.smith@company.com tells the system to look for a user record for John Smith.  
• Type: string  
• Default value: period (. ) |
| glide.email.notification.save_when_no_recipients | Controls whether a notification-generated sys_mail record is saved even if there are no recipients. Used in conjunction with other notification recipient logging properties, this property enables troubleshooting problems with notifications.  
• Type: true | false  
• Default value: true |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.outbound.header.auto_submitted</td>
<td>Stores the value used in the &quot;Auto-submitted&quot; outbound email header. Clear the property value to remove the &quot;Auto-submitted&quot; header from all outbound emails. Some spam filters flag auto-generated email as spam.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: auto-generated</td>
</tr>
<tr>
<td>glide.email.outbound.max_attachment_count</td>
<td>Sets the maximum number of attachments allowed per outbound email. This property is available starting with the Eureka Patch 4 release.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 30</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.email.outbound.max_body_bytes</td>
<td>Sets the maximum body size in bytes allowed per outbound email.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1048576</td>
</tr>
<tr>
<td>glide.email.outbound.max_total_attachment_size_bytes</td>
<td>Sets the maximum total attachment size in bytes allowed per outbound email. This property is available starting with the Eureka Patch 4 release.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 18874368</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.email.override.url</td>
<td>Sets the URL to use in emailed links in place of the instance URL. The URL should end with nav_to.do. An example value is: <a href="https://servicenow.customerdomain.com/production/nav_to.do">https://servicenow.customerdomain.com/production/nav_to.do</a>. This property is suitable for customers who use custom redirect URLs for their instances.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: instance URL</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.cms.use_email_override_url | Forces the system to use the glide.email.override.url property, rather than the glide.servlet.uri property, when a notification has a link to a CMS page in an instance.  
  • Type: true | false  
  • Default value: false  
  • Location: System Property [sys_properties] table |
| glide.email.smtp.max_recipients | Specifies the maximum number of recipients the instance can list in the To: line for a single email notification. Notifications that would exceed this limit instead create duplicate email notifications addressed to a subset of the recipient list. Each email notification has the same maximum number of recipients.  
  • Type: integer  
  • Default value: 100 |
| glide.email.smtp.max_send | Specifies how many emails to send through each new SMTP connection. The instance establishes a new SMTP connection if there are more emails to send than the specified value.  
  • Type: integer  
  • Default value: 100 |
| glide.email.text_plain.strip_xhtml | Indicates whether both outbound and inbound emails that are shown in comments convert the XML to plain text (true) or preserve the XML (false).  
  • Type: true | false  
  • Default value: true |
| glide.imap.secure | Specifies whether to enable SSL encryption for connections to the IMAP server.  
  • Type: true | false  
  • Default value: false |
| glide.imap.secure.port | Specifies the communications port for IMAP secure connections.  
  • Type: string  
  • Default value: 995 |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.imap.tls | Specifies whether to start the IMAP server in Transport Layer Security (TLS) mode.  
- Type: true | false  
- Default value: false |
| glide.notification.recipient.exclude_logging | Master switch to enable or disable logging all reasons a recipient was excluded. If set to true, the subsequent properties dealing with the exclusion of logging are enabled. If it is set to false, none of the subsequent properties relating to the exclusion of logging are enabled. This property cannot suppress log messages generated by the glide.email.test.user property.  
- Type: true | false  
- Default value: true |
| glide.notification.recipient.exclude_logging.device_inactive | Logs recipients who are excluded because their chosen notification device record is marked as inactive.  
- Type: true | false  
- Default value: true |
| glide.notification.recipient.exclude_logging.device_schedule | Logs recipients who are excluded based on the Schedule field on the New Device for System Administrator form for their chosen notification device.  
- Type: true | false  
- Default value: true |
| glide.notification.recipient.exclude_logging.event_creator | Logs recipients who are excluded because they initiated the notification event, such as updating an incident record, and the Send to Event Creator check box is cleared on the notification record.  
- Type: true | false  
- Default value: true |
| glide.notification.recipient.exclude_logging.invalid_email | Logs recipients who are excluded because the email address for that user is invalid, for example the @ is missing, or empty.  
- Type: true | false  
- Default value: true |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.notification.recipient.exclude_logging.user_calendar_integration_disabled | Logs recipients of calendar invitations who are excluded because the **Calendar Integration** field is set to **None** on the user record.  
  • Type: true | false  
  • Default value: true |
| glide.notification.recipient.exclude_logging.user_inactive | Logs recipients who are excluded because the **Active** check box is cleared on the user record.  
  • Type: true | false  
  • Default value: true |
| glide.notification.recipient.exclude_logging.user_notification_disabled | Logs recipients who are excluded because the **Notification** field is set to **Disabled** on the user record.  
  • Type: true | false  
  • Default value: true |
| glide.notification.recipient.include_logging | Master switch to enable or disable logging all reasons a recipient was included. If set to true, the subsequent properties dealing with the inclusion of logging are enabled. If it is set to false, none of the subsequent properties relating to the inclusion of logging are enabled.  
  • Type: true | false  
  • Default value: true |
| glide.notification.recipient.include_logging.delegate | Logs recipients who are included because they are delegates of another user.  
  • Type: true | false  
  • Default value: true |
| glide.notification.recipient.include_logging.event_parm | Logs recipients who are included because they are in the **parm1** or **parm2** fields of the event record.  
  • Type: true | false  
  • Default value: true |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| glide.notification.recipient.include_logging.recipient_fields | Logs recipients who are included via a notification target record, such as an incident record, specified in the **Users/Groups in Field** field for the notification record. The recipient_fields are fields in the target record that contain a recipient to add. For example, if the record that triggered the notification is an incident, and the **assigned_to** field for the incident is listed in recipient_fields, that user is included as a recipient.  
  - Type: true | false  
  - Default value: true |
| glide.notification.recipient.include_logging.recipient_groups.group_email | Logs recipients who are included in a group email for any group provided in the notification record's recipient_groups or the event **parm1** or **parm2** field.  
  - Type: true | false  
  - Default value: true |
| glide.notification.recipient.include_logging.recipient_groups.manager | Logs recipients who are included because they manage any group provided in the notification record's recipient_groups or the event **parm1** or **parm2** field.  
  - Type: true | false  
  - Default value: true |
| glide.notification.recipient.include_logging.recipient_groups.membership | Logs recipients who are included via membership in any group provided in the notification record recipient_groups or the event **parm1** or **parm2** field.  
  - Type: true | false  
  - Default value: true |
| glide.notification.recipient.include_logging.recipient_users | Logs recipients who are included via notification record's **Users** field (recipient_users).  
  - Type: true | false  
  - Default value: true |
| glide.notification.recipient.include_logging.subscription | Logs recipients because they are subscribed via User Notification Preferences.  
  - Type: true | false  
  - Default value: true |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.pop3.parse_start</td>
<td>Specifies the text that indicates the beginning of the email body section. The instance parses name:value pairs within this section to set or update field values when processing inbound email actions.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td>glide.pop3.parse_end</td>
<td>Text indicating the end of the email body section where the instance should parse name:value pairs to update field values when processing inbound email actions.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: none</td>
</tr>
<tr>
<td>glide.smtp.dateformat</td>
<td>Specify the date format to use for outgoing email notifications.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: date format listed in email sender's user record</td>
</tr>
<tr>
<td></td>
<td>[sys_user.date_format]</td>
</tr>
<tr>
<td>glide.smtp.precedence_bulk</td>
<td>Specifies whether outbound email includes the header &quot;Precedence: bulk&quot;. Some spam filters flag bulk email as spam. Set the value to false to remove this header from outbound email.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td>glide.smtp.secure</td>
<td>Use the glide.smtp.encryption property in the Outgoing Mail Server section of the Mail Properties page to specify how to encrypt communications with the SMTP server.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td>glide.smtp.timeformat</td>
<td>Specify the time format to use for outgoing email notifications.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: time format listed in email sender's user record</td>
</tr>
<tr>
<td></td>
<td>[sys_user.time_format]</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.smtp.tls</td>
<td>Use the glide.smtp.encryption property in the Outgoing Mail Server section of the Mail Properties page to specify how to encrypt communications with the SMTP server.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td>glide.ui.activity.email.use_display</td>
<td>Specifies whether to display email addresses or user IDs (display value from the User table) in email headers. If true, the instance searches for a user record with a matching email address. If it cannot find a matching user record, it displays the email address.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td>glide.ui.email_client.email_address.disambiguator</td>
<td>Sets the columns from the User [sys_user] table that the autocomplete list displays. Separate each column name with a semicolon character (;).</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: name</td>
</tr>
<tr>
<td></td>
<td>• Learn more: Displaying Additional Information in the Email Client Autocomplete</td>
</tr>
<tr>
<td>glide.ui.incident_activity.max_addresses</td>
<td>Specifies the maximum number of addresses to list in an email audit record. If the number of addresses exceeds this limit, the instance truncates the list after the maximum value and displays an ellipsis character (...).</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5</td>
</tr>
<tr>
<td>NotifyAffectedCI.max_rel_level</td>
<td>Sets a value used by the Affected ci notifications business rule, which notifies subscribers when configuration items (CIs) are affected by tasks. The business rule generates notifications for parent CIs up to the level defined by this property. You might need to adjust the property value according to the complexity and depth of your CI relationships.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 5</td>
</tr>
</tbody>
</table>

**Deprecated mail server properties**

In previous releases, you could configure email accounts from the Email Properties page.
Starting with the Geneva release, you must configure email accounts from the **System Mailboxes > Email Accounts** module. Properties related to email accounts are deprecated. The same functionality in these deprecated properties is configured on the Email Account form.

**Properties deprecated in the Geneva release**

- SMTP mail server (glide.email.server)
- SMTP account (glide.email.user)
- Outgoing mail display name (glide.email.username)
- SMTP account password (glide.email.user_password)
- POP3 mail server (glide.pop3.server)
- POP3 account (glide.pop3.user)
- POP3 account password (glide.pop3.password)
- POP3 server port (110) (glide.pop3.port)
- Connect to POP3 server using SSL encryption (glide.pop3.secure)
- SMTP server requires username and password authentication (glide.smtp.auth)
- SMTP server port (25) (glide.smtp.port)
- SMTP Encryption (glide.smtp.encryption)
- [Legacy] TLS security for SMTP (glide.smtp.tls)

**Email diagnostics**

The Diagnostics and Connection page displays information about the current state of your email configuration.

Email diagnostics are available from either of these modules:

- **System Mailboxes > Email Diagnostics**
- **System Diagnostics > Email Diagnostics**
Figure 764: Email diagnostics

Table 701: Mail diagnostics

<table>
<thead>
<tr>
<th>Diagnostic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email Sending is [Status]</td>
<td>Status of outbound email as either Enabled or Disabled.</td>
</tr>
<tr>
<td>Diagnostic</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Email in Queue</td>
<td>Number of email messages in the Outbox.</td>
</tr>
<tr>
<td>Last Sent Mail</td>
<td>Date and time the last email message was sent.</td>
</tr>
<tr>
<td>SMTP Sender State</td>
<td>Current state of the SMTP Sender job, which determines how often to send email. By default, this job runs every minute. If the state is anything other than Ready, the instance may not be able to send email.</td>
</tr>
<tr>
<td>SMTP Sender Processing Time</td>
<td>Duration of the last SMTP Sender job run. This value should be shorter than the SMTP Sender interval.</td>
</tr>
<tr>
<td>SMTP Sender Job Last Run</td>
<td>Date and time when the SMTP Sender job last ran.</td>
</tr>
<tr>
<td>Default SMTP Status</td>
<td>Indication of whether the SMTP connection was successful, shown only if the email accounts feature is active. Click Default SMTP to change your SMTP account settings.</td>
</tr>
<tr>
<td>SMS Sender State</td>
<td>Current state of the SMS Sender job, which determines how often to send SMS notifications. By default, this job runs every minute. If the state is anything other than Ready, the instance may not be able to send SMS notifications. This diagnostic is available starting with the Fuji release.</td>
</tr>
<tr>
<td>SMS Sender Processing Time</td>
<td>Duration of the last SMS Sender job run. This value should be shorter than the SMS Sender interval. This diagnostic is available starting with the Fuji release.</td>
</tr>
<tr>
<td>SMS Sender Job Last Run</td>
<td>Date and time when the SMS Sender job last ran. This diagnostic is available starting with the Fuji release.</td>
</tr>
<tr>
<td>Email Receiving is [Status]</td>
<td>Status of inbound email as either Enabled or Disabled.</td>
</tr>
<tr>
<td>Email Receiving</td>
<td>Date and time the last email message was received.</td>
</tr>
<tr>
<td>Email Reader Status</td>
<td>Current status of the email reader job, which downloads any email waiting on the mail server and creates email.read events.</td>
</tr>
<tr>
<td>Email Reader Processing Time</td>
<td>Duration of the last reader job run. This value should be shorter than the reader interval.</td>
</tr>
<tr>
<td>Job Last Run</td>
<td>Date and time when the reader job last ran.</td>
</tr>
</tbody>
</table>

**Connection Status**
### Email logs

The email log records all email notifications sent from all instances within the system.

This is a verbose and unfiltered view of email. For a more detailed view, see the [System Mailbox application](#).

This log provides the following information for all notifications.

#### Table 702: Email log

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox</td>
<td>The system mailbox to use for filtering the email notifications displayed.</td>
</tr>
<tr>
<td>State</td>
<td>The current state of the notification (Error, Ignored, Processed, or Ready).</td>
</tr>
<tr>
<td>Receive type</td>
<td>The type of inbound email notification (None, Forward, New, or Reply).</td>
</tr>
<tr>
<td>Type</td>
<td>The status of the email notification. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• received: The server received this email.</td>
</tr>
<tr>
<td></td>
<td>• received - ignored: The server received this email, but it was ignored by</td>
</tr>
<tr>
<td></td>
<td>the instance for inbound email action purposes. Typically, these emails</td>
</tr>
<tr>
<td></td>
<td>are either spam or auto-replies. See the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - failed: The server has attempted to send the email and failed. See</td>
</tr>
<tr>
<td></td>
<td>the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - ignored: The server skipped sending this email. Typically, this is</td>
</tr>
<tr>
<td></td>
<td>for an email which was generated but lacked a recipient email address or</td>
</tr>
<tr>
<td></td>
<td>is a duplicate email. See the Error String field for details.</td>
</tr>
<tr>
<td></td>
<td>• send - ready: The email is ready to be sent, but has not been sent out by</td>
</tr>
<tr>
<td></td>
<td>the mail server. Typically, an email remains in this state for only a</td>
</tr>
<tr>
<td></td>
<td>short time.</td>
</tr>
<tr>
<td></td>
<td>• sent: The email was sent by the instance without any errors or issues.</td>
</tr>
<tr>
<td>Target</td>
<td>A Document ID reference to the record if the email is generated by an insert,</td>
</tr>
<tr>
<td></td>
<td>update, or delete of a particular record.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User</td>
<td>The name of the user, from the user record, of the instance from which the email notification was sent.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This is a string field.</td>
</tr>
<tr>
<td>Notification Type</td>
<td>The type of notification. Choices are:</td>
</tr>
<tr>
<td></td>
<td>• None</td>
</tr>
<tr>
<td></td>
<td>• SMS</td>
</tr>
<tr>
<td></td>
<td>• SMTP</td>
</tr>
<tr>
<td>UID</td>
<td>The unique ID for the server.</td>
</tr>
<tr>
<td>Created</td>
<td>The date and time of the email activity for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Deleted</td>
<td>An indication of whether the email was deleted from an instance mailbox.</td>
</tr>
<tr>
<td>Weight</td>
<td>The weight of the email, which determines the sending priority relative to other notifications on the same table.</td>
</tr>
<tr>
<td>Importance</td>
<td>An indication that the email was sent with a changed level of importance, such as Urgent.</td>
</tr>
<tr>
<td>Originating Event and Notification</td>
<td>An embedded list that stores the event and notification that initiated the email notification. For more information, see <a href="#">Events</a> on page 2996.</td>
</tr>
<tr>
<td>Subject</td>
<td>A configured description of the action that generated the email notification. You create the subject text for notifications in <strong>System Notification &gt; Email &gt; Notifications</strong>.</td>
</tr>
<tr>
<td>Error String</td>
<td>The error string captured from the email server to determine why the email was not sent. This is logged only if the email is send-failed.</td>
</tr>
<tr>
<td>Recipients</td>
<td>The email address of the recipient of each notification.</td>
</tr>
<tr>
<td>Body</td>
<td>The body of the email, displayed in raw HTML markup. Use the related link Preview HTML Body to see the body text as rendered HTML.</td>
</tr>
<tr>
<td>Content type</td>
<td>The email content type.</td>
</tr>
<tr>
<td>Headers</td>
<td>Any headers embedded in the email.</td>
</tr>
</tbody>
</table>

Invalid email addresses that the instance strips out of outbound email messages are logged, starting with the Geneva release.

*Email size limits*
To prevent issues with large email messages, the system enforces configured limits on the maximum allowed email body size, total attachment file size, and number of attachments per email.
Administrators can modify the maximum values using properties.

**Note:** The system cannot send or receive emails that exceed the maximum email size, even if one or more properties are configured to allow bigger attachments.

Message body size limit properties
Several properties enforce the maximum email body size allowed for inbound and outbound emails.

**Properties**

In some situations, an email message may contain more information in the body than the system can process, especially when string searches and regular expressions are used. By default, the system can process only up to 16MB worth of information in the email body and body_text fields (8MB each). If the email body exceeds this limit, the system truncates the body text.

Users with the admin role can configure the following properties to control email body size limits. The properties can be added to the System Property [sys_properties] table. Neither property should exceed the system limit of 16MB. Setting either of the following properties to an excessively large value may cause performance issues.

**Table 703: Message body size limit properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.glide.email.max_body_bytes</td>
<td>Sets the maximum body size in bytes allowed per inbound email. Ensure that the sum of this property and the glide.email.inbound.max_total_attachment_size_bytes property is well below the maximum total email size.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> 1048576</td>
</tr>
<tr>
<td></td>
<td>• <strong>Learn more:</strong> Inbound Email Attachment Processing</td>
</tr>
<tr>
<td>glide.email.outbound.max_body_bytes</td>
<td>Sets the maximum body size in bytes allowed per outbound email. Ensure that the sum of this property and the glide.email.outbound.max_total_attachment_size_bytes property is well below the maximum total email size.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> integer</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default value:</strong> 1048576</td>
</tr>
<tr>
<td></td>
<td>• <strong>Learn more:</strong> Outbound Email Attachment Processing</td>
</tr>
</tbody>
</table>

**Note:** A different property, com.glide.attachment.max_size, sets the maximum file size allowed for any attachment in the system and overrides any larger values of glide.email.inbound.max_total_attachment_size_bytes and glide.email.outbound.max_total_attachment_size_bytes.
**Inbound email body processing**

For inbound emails, the system enforces the maximum body size as set by the `com.glide.email.max_body_bytes` property. When the body size exceeds the configured value:

- The system does not run inbound email actions that would otherwise be triggered by the email.
- The system truncates the Body text on the Email form.
- The system logs a warning and sets the Error string field on the Email form. The log message for such an email might look like this:

```
Email set to receive-ignored because its size exceeds the value set in com.glide.email.max_body_bytes. 1995 character(s) were truncated from the body field.
```

**Outbound email body processing**

For outbound emails, the system enforces the maximum body size as set by the `glide.email.outbound.max_body_bytes` property. When the body size exceeds the configured value:

- The system does not send the email.
- The system truncates the Body text on the Email form.
- The system logs a warning and sets the Error string field on the Email form. The log message for such an email might look like this:

```
Email set to send-ignored because its size exceeds the value set in glide.email.outbound.max_body_bytes. 1337 character(s) were truncated from the body field.
```

**Attachment limit properties**

Several properties control email attachment limits.

**Properties**

All the properties are located in the System Property [sys_properties] table. Setting any of the following properties to an excessively large value may cause performance issues.

**Table 704: Attachment limit properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>glide.email.inbound.max_attachment_count</code></td>
<td>Sets the maximum number of attachments allowed per inbound email.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Type</strong>: integer</td>
</tr>
<tr>
<td></td>
<td>- <strong>Default value</strong>: 30</td>
</tr>
<tr>
<td></td>
<td>- <strong>Learn more</strong>: Inbound Email Attachment Processing</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| glide.email.inbound.max_total_attachment_size_bytes | Sets the maximum total attachment size in bytes allowed per inbound email.  
  - **Type**: integer  
  - **Default value**: 18874368  
  - **Learn more**: Inbound Email Attachment Processing |
| glide.email.outbound.max_attachment_count | Sets the maximum number of attachments allowed per outbound email.  
  - **Type**: integer  
  - **Default value**: 30  
  - **Learn more**: Outbound Email Attachment Processing |
| glide.email.outbound.max_total_attachment_size_bytes | Sets the maximum total attachment size in bytes allowed per outbound email. Note that in order to send an email, the system must encode the contents of the email. This process may significantly increase the size of the email, including any attachments. It is best to set this property to a value well below the maximum email size.  
  - **Type**: integer  
  - **Default value**: 18874368  
  - **Learn more**: Outbound Email Attachment Processing |

**Note:** A different property, com.glide.attachment.max_size, sets the maximum file size allowed for any attachment in the system and overrides any larger values of glide.email.inbound.max_total_attachment_size_bytes and glide.email.outbound.max_total_attachment_size_bytes.

**Inbound email attachment processing**

For inbound emails, the system enforces the maximum number and size of attachments as set by the glide.email.inbound.max_attachment_count and glide.email.inbound.max_total_attachment_size_bytes properties. When the attachments for an inbound email exceed either value, the system logs a warning and discards the excess attachments. The order in which the system processes the attachments determines which attachments are discarded. This order may not be consistent from email to email.

**Inbound email attachment processing**

For outbound emails, the system enforces the maximum number and size of attachments as set by the glide.email.outbound.max_attachment_count and glide.email.outbound.max_total_attachment_size_bytes properties. Email records are created from a number of sources and may exceed the configured attachment limits.
Emails that are ready to be sent from the Email [sys_email] table are subject to the outbound attachment limits. Emails that exceed either limit trigger a warning in the email system log and are sent with attachments up to the maximum number or total file size.

The log message for such an email might look like this:

Maximum combined attachment size exceeded. (max:15728640 bytes). One or more attachment records ignored.

**Emails for notifications, scheduled reports, and exported tables**

Notifications can be set to include all the attachments from the record that triggers the notification. If the attachments exceed either of the outbound email attachment limits, the system excludes the excess attachments from the email and logs a warning message.

Reports can be scheduled for email distribution as attachments. Large reports may exceed the outbound attachment size limit. In this case the system sends the scheduled report email without the report attached and logs a warning message. To avoid the issue, send links to large reports instead of sending the reports as attachments.

If a user attempts to export a number of records from a list that exceeds a configured warning threshold, a dialog box offers the option to email the exported records as an attachment. If the attachment exceeds the outbound attachment size limit, the system sends the email without the exported record list attached and logs a warning message.

**Maximum email body size**

A system property controls the maximum size of a message body before the instance stops processing inbound email actions.

In some situations, an email message may contain more information in the body than the instance's data policy permits a field to write to the database. By default, an inbound email action can process only up to 16MB worth of information in the message body. If the email body exceeds this limit, the instance truncates the message body. The default value is set to match the data policy write limit of 16MB. You cannot set a higher value without conflicting with the write data policy limit. If you want to set a lower limit, you can add this system property.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>com.glide.email.max_body_bytes</td>
</tr>
<tr>
<td>Description</td>
<td>Specify the maximum size of an email message body in bytes. The size cannot exceed the default value of 16MB.</td>
</tr>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Value</td>
<td>16777216</td>
</tr>
</tbody>
</table>

**Note:** Email attachments have a separate size limit.

**Next steps after enabling email**

After enabling email on your instance, consider performing several of these important tasks.

- Test the email configuration by routing all email to a single user (set the glide.email.test.user property).
- Review the baseline email notifications and templates to determine if they meet your business needs.
• Review the baseline inbound email actions to determine if they meet your business needs.
• Determine if you want to use email filters to restrict the email the instance receives.
• Determine if you want to implement a retention policy to archive and destroy email at certain intervals.
• Determine what kind of watermarks outbound email uses to associate records with email messages.
• Determine if you want to create users when the instance receives an email from an unrecognized user.
• Set the precedence of outgoing mail. By default, the instance sends email with a precedence of bulk (set the glide.smtp.precedence_bulk property).
• Consider preventing untrusted users from triggering inbound actions to prevent unwanted email from affecting your instance.
• Implement a spam filter to restrict unwanted messages sent to your custom email addresses.

Create an email notification

Creating an email notification involves specifying when to send it, who receives it, and what it contains.

Role required: admin

Note: If you do not see all the fields on the form, switch to the Advanced view.

1. Configure email properties to enable your instance to send and receive email.
2. Navigate to System Notification > Email > Notifications.
3. Click New.
4. Fill in the fields at the top of the Email Notifications form, as appropriate (see table).
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the email notification. Descriptive names help identify the purpose of the email notification. For example, Incident Opened &amp; Unassigned.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the database table to link the notification to, for example Incident [incident].</td>
</tr>
<tr>
<td><strong>Attention:</strong></td>
<td>Do not select the Task [task] table. This table is for extending other tables. Notifications that run on the Task table directly are not supported.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Only tables and database views that are in the same application scope appear in the list.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the type of notification you are creating: EMAIL or Meeting Invitation.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to enable the email notification.</td>
</tr>
<tr>
<td>Description</td>
<td>Type a description for this notification.</td>
</tr>
</tbody>
</table>

5. Fill in the fields on the **When to send** tab (see table).
Note: If the same trigger generates multiple notifications, the system only sends one notification. The system considers all other notifications, even if they have a different subject and body, as duplicates. The Ignore Duplicates business rule controls this functionality.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send when</td>
<td>Select under what condition the notification is sent: when a record is inserted or updated or when a particular event is triggered.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Weight</td>
<td>[Required] Set a numerical value for the notification priority relative to other notifications with the same target table and recipients. The system only sends the notification with the highest weight. All other notifications are moved from the Outbox to the Skipped mailbox. The default value 0 causes the system to always send the notification (assuming the conditions are met). For example, suppose that a service desk agent adds a comment to an incident and shortly thereafter closes it. By default, these actions trigger both the Incident commented and Incident Closed notifications. However, both notifications are from the Incident table and also notify the incident caller. The system only sends the notification with the highest weight, which in this case is the Incident Closed notification. <strong>Note:</strong> The SMTP Sender scheduled job determines how often to send email. By default, this job runs every minute.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Use the condition builder to select the conditions under which this notification is sent. For example, select <strong>Priority &gt; greater than &gt; 3 - Moderate</strong> to send the notification only for High and Critical priority incidents.</td>
</tr>
<tr>
<td>Inserted</td>
<td>Select the check box to enable email notification when a record is inserted. This field is visible only when the Send when field has been set to <strong>Record inserted or updated.</strong></td>
</tr>
<tr>
<td>Updated</td>
<td>Select the check box to enable email notification when a record is updated. This field is visible only when the Send when field has been set to <strong>Record inserted or updated.</strong></td>
</tr>
<tr>
<td>Event name</td>
<td>Select the event that triggers this notification. This field is visible only when the Send when field has been set to <strong>Event is fired.</strong></td>
</tr>
</tbody>
</table>
6. Fill in the fields on the **Who will receive** tab (see table).

**Tip:** Consider limiting the recipient list of any notification to **1000** users. By default, if a notification has more than 100 intended recipients. The system creates multiple notification messages with up to 100 recipients each. If you want to change the recipient limit, set the system property glide.email.smtp.max_recipients.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Advanced condition         | Create a script to perform certain actions, like sending a notification based on the current email record, changing field values, or changing system properties. **The advanced condition** script must return **true** or set a global answer variable to **true** to send the notification. The advanced condition script uses the following global variables:  
  • current: contains the current record from the table to which the notification is linked.  
  • event: contains the event that triggered the notification.  
  **Note:** The Advanced condition field is evaluated in addition to other conditions you set on the notification. Both the condition and advanced condition must evaluate to true to send the notification. |

Notifications can be sent to specific Users and Groups or to User/Groups in fields on the record that generated this notification.

<table>
<thead>
<tr>
<th>Users</th>
<th>Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users/Groups in fields</td>
<td>Assignment group</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Users</td>
<td>Select the users you want to receive the email notification. You can search for users with the reference lookup icon or manually add their email addresses. This list of users is static.</td>
</tr>
<tr>
<td>Users/groups in fields</td>
<td>Select users or groups from reference fields. For example, if a notification uses the Incident [incident] table, then you can select users or groups from incident fields like <em>Opened by</em> and <em>Assignment group</em>. This list of users or groups is variable and depends upon the values of the associated task record.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can dot-walk to values in reference fields by clicking the plus sign in the field selector and then selecting the related field.</td>
</tr>
<tr>
<td>Groups</td>
<td>Select the groups you want to receive the email notification. You can search for groups with the reference lookup icon or by manually entering the group name. This list of groups is static.</td>
</tr>
<tr>
<td>Exclude delegates</td>
<td>Select this option to prevent the instance from sending email notifications to delegates of the users you selected.</td>
</tr>
<tr>
<td>Send to event creator</td>
<td>Select this check box to send the notification to the person who performed the action that started the notification process if the person is also a recipient. If the event creator is not specified in one of the recipient fields, the event creator does not receive a notification regardless of the setting in this field.</td>
</tr>
<tr>
<td></td>
<td>For new notifications, this option is selected by default.</td>
</tr>
<tr>
<td></td>
<td>If you want know why you may not be receiving certain email notifications, see the blog post <em>Troubleshooting email notifications - Send to the Event Creator</em> by a ServiceNow Technical Support Engineer in the ServiceNow Community.</td>
</tr>
<tr>
<td>Event parm 1 contains recipient</td>
<td>Select this check box if the event parameter 1 contains one or more email recipients (in a comma separated-list). This field is visible only when the <em>Send when</em> field is set to <em>Event is fired</em>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Event parm 2 contains recipient</td>
<td>Select this check box if the event parameter 2 contains one or more email recipients (in a comma-separated list). This field is visible only when the <strong>Send when</strong> field is set to <strong>Event is fired</strong>.</td>
</tr>
<tr>
<td>Subscribable</td>
<td>Select this check box to allow all users to subscribe to this notification. See <strong>Subscription-based notifications</strong> on page 2896 for more information.</td>
</tr>
</tbody>
</table>

**Note:** Do not enable this option for notifications that contain sensitive or protected data or where you want to restrict who can see it.

**Note:** The system does not exclude recipients based on access controls. Recipients can receive email about records that they cannot normally access from the user interface. For example, requesters can receive email about incidents and catalog requests opened on their behalf even though they normally do not have access to these records. If a notification includes record details, verify that all recipients need these details. If the record contains sensitive or protected data, consider restricting the recipient list to just those users and groups who normally have access to it, and do not enable the **Subscribable** option.

7. Fill in the fields on the **What it will contain** tab (see table).
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email template</td>
<td>If you want to reuse existing content, select an email template to add content to the email notification.</td>
</tr>
<tr>
<td>Subject</td>
<td>Enter the subject line for the email message. The subject can include variables from the <code>Select variables</code> column.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Message HTML</td>
<td>Enter the content of the email notification message. The message can include variables from the Select variables column. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table such as an incident short description or comments and work notes. The <strong>Message HTML</strong> field is visible only if you set the content type to <strong>HTML and plain text</strong> or <strong>HTML only</strong>. To prevent adding extra <code>&lt;p&gt;</code> and <code>&lt;div&gt;</code> elements to your email notifications, see the blog post <em>Extra line spacing with paragraph tags in email client</em> by a ServiceNow employee in the ServiceNow Community.</td>
</tr>
<tr>
<td>SMS alternate</td>
<td>Enter the notification message to send to an SMS device. The SMS alternate message is limited to 140 characters.</td>
</tr>
<tr>
<td>Importance</td>
<td>Set the importance of the email message to low or high.</td>
</tr>
<tr>
<td>Content type</td>
<td>Select the content type for the email notification: • <strong>HTML and plain text</strong> • <strong>HTML only</strong> • <strong>Plain text only</strong> By default, <strong>HTML only</strong> is enabled.</td>
</tr>
<tr>
<td>Include attachments</td>
<td>Select this check box to send all attachments from the triggering record as email attachments.</td>
</tr>
<tr>
<td>Omit watermark</td>
<td>Use this check box to apply or remove the watermark attached to each email. Do not remove the watermark as a reply to an email without a watermark creates an incident rather than updating the incident the original email referred to. For more information, and an alternative way to hide watermarks, see <em>Watermarks on notification emails</em> on page 2855.</td>
</tr>
<tr>
<td>Message Text</td>
<td>Enter the notification message to send in plain text. This field is visible only if you set the content type to <strong>HTML and plain text</strong> or <strong>Plain text only</strong>.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Enter the email address that you want the email notification to use in the From field. For example, <a href="mailto:helpdesk@yourcompany.com">helpdesk@yourcompany.com</a>. The email must be in a valid format, otherwise a notification message appears near the field.</td>
</tr>
<tr>
<td>Reply to</td>
<td>Enter the email address you want people to use when replying to the email notification. For example, <a href="mailto:helpdesk@yourcompany.com">helpdesk@yourcompany.com</a>. The email must be in a valid format, otherwise a notification message appears near the field.</td>
</tr>
<tr>
<td>Push message only</td>
<td>Select this option to send this notification only as a push notification to a mobile device. The Push Notification feature must be active.</td>
</tr>
<tr>
<td>Push messages</td>
<td>Associate one or more push messages with this notification. The Push Notification feature must be active.</td>
</tr>
</tbody>
</table>

8. Click **Submit**.

Images can be inserted into email notifications that were created in the Eureka release or that were converted to rich format using the HTML editor. The images can be stored in the image library in your instance, or they can be inserted as attachments.

*Convert legacy email notifications to rich HTML*

By default, new email notifications are created in the rich HTML format. But you can also convert legacy notifications to rich HTML.

Role required: admin

1. Navigate to **System Policy > Email > Notifications**.
2. On the **Email Notifications** list screen, click the name of the email notification you want to convert.
3. Click the **What it will contain** tab.
4. Click **Switch to Rich HTML Editor**.

Any raw HTML in the **Message** field is rendered as WYSIWYG text. Additionally, any mail scripts in the body are automatically saved to the Email Script [sys_script_email] table and are replaced in the notification body with an embedded script tag. This makes the notification body easier to read.
When you convert an email notification that was created in a version prior to Eureka to rich HTML, mail scripts are automatically moved to the Email Script [sys_script_email] table and an embedded script tag with the name of the script is automatically inserted into the body of the notification.

When creating new email notifications in the Eureka release, it is a best practice to write mail scripts using System Policy > Email Notification > Email Scripts. When the scripts are completed, add a `${mail_script:script name}` embedded script tag to the email notification body. This makes it easy to use the same scripts in multiple email notifications. All you need to copy and paste from one notification to the next is the embedded script tag.

If you manually enter a mail script, any text bounded by `<mail_script> </mail_script>` in the body of a new or converted email notification or template which is saved to the record, a message asks whether the mail script should be converted.
In many cases, an unconverted mail script fails to run from inside the HTML editor. If you select Yes, the script is added to the Email Script [sys_script_email] table and is automatically replaced in the body with an embedded script tag. You can view the mail scripts in their original form by opening the email notification and clicking the Show Notification Scripts related link.

**Advanced conditions for email notifications**

Use an advanced condition to send a notification based on the current email record, changing field values, or system properties.

The advanced condition script must set the answer variable to true to send the notification. For example, to prevent the system from sending an email notification if the sender of a self-service request is a member of the XYZ group, use this code:

```javascript
var groupMember = gs.getUser();
if(groupMember.isMemberOf('XYZ')){
    answer = false;
} else{
    answer = true;
}
```

You can add a script-based condition in the Advanced condition field by configuring the Email Notification form and adding the field. You can access the field in the Advanced view without configuring the form.

The advanced condition script uses the following business rule global variables:

- **current**: contains the current record from the table to which the notification is linked.
- **event**: contains the event that triggered the notification.

**Note**: The Advanced condition field is evaluated in addition to other conditions you set on the notification. Both the Condition and Advanced condition must evaluate to true in order to send the notification.

**Edit HTML content in an email notification**

For added control over the content of a converted email notification, you can edit the underlying HTML.

Role required: admin

1. With the converted email notification displayed in the Message text field, click the HTML button in the rich HTML editor.
2. Make the needed changes to the HTML.
3. Click **Update**.

*Document attachments on an email notification*

You can attach documents and reports to email notifications by scripting or linking to the sys ID of the record.

You can include all attachments from the source record with the notification. For example, if an incident update generates a notification, you can include all attachments from the incident record with the notification. To include all attachments from the source record, select the check box for the **include attachments** field. Note that email messages, including attachments, cannot exceed the maximum email size. This size includes MIME encoding, which increases total attachment size by approximately 37%.

**Attaching documents with scripting**

Using scripting, you can attach documents by linking to them, or you can attach various types of reports by specifying their IDs in the system.
**Linking to an attachment**

You can add attachments to a notification by linking to the attachment record in the message of the notification. Linking to attachment records in this fashion requires using email notification scripting. For example:

```javascript
template.print ( 'Attachment: <a href="/sys_attachment.do?sys_id=' + gr. sys_id + '">' + gr. file_name + '</a>
 ' ) ;
```

**Attaching reports using the Sys ID**

You can also attach various types of reports, including gauges, dashboards, and charts, to a notification. The scripts to attach these reports take the following syntax:

```javascript
${report:X:Y}
```

where:

- X is the type of report you want to attach (reportID, gaugeID, dashboardID, or chartID).
- Y is the sys ID of the report, gauge, dashboard, or chart to be attached.

For example:

- `${report:reportID:<abc123>}`
- `${report:gaugeID:<abc123>}`
- `${report:dashboardID:<abc123>}`
- `${report:chartID:<abc123>}`

*Line breaks in email notifications and rich HTML*

Rich HTML provides additional control over line breaks in your email notifications and templates.

To accommodate this, a **Newlines to HTML** check box is available in the **Email Script** form.
Selecting the **Newlines to HTML** check box indicates that the method for handling line breaks in earlier versions carries forward for email notifications and templates. When an email notification or template is converted to rich HTML, the **Newlines to HTML** check box is automatically selected.

For new mail scripts, it is a best practice to add correct HTML line breaks to template.print() statements.

If an email notification or template to rich HTML is not converted to rich HTML, newlines are automatically wrapped with `<div>` tags, the same as previous versions. The old mail scripts still work; however, the administrator does not enjoy the benefits of working in the rich HTML format, and does not have as much control over exact HTML formatting.

### Best Practice

When writing new scripts, it is best to insert explicit HTML line breaks and clear the **Newlines to HTML** check box so that no HTML tags are injected when email notifications are generated. The same approach is recommended for existing notifications and templates. That is, a best practice would be to replace `template.print("\n")` JavaScript function calls with `template.print("<br />")`. This gives you better control over the HTML formatting of your email notifications.

#### Preview email notifications

You can preview what notifications look like before you actually enable the instance to send them.

Role required: admin

---

**Note:** If you are using Internet Explorer, you must have version 9 or later to use this feature.

You can preview both types of notifications as specified by the **Send when** field on the Notification form:

- **Record inserted or updated**: A change to record in the instance triggers the notification.
- **Event is fired**: An event, such as expiration of a certificate or an inbound email action, triggers the notification.

1. Navigate to **System Notification > Email > Notifications**.
2. Open the notification or create one. You must save the record before you can view the preview accurately.

3. Click the **Preview Notification** UI action on the top of the form. The Notification Preview window opens.

4. You can see the content of the notification, modify the preview records to see what changes take place in the notification content, and view who can receive the notification (see table for field descriptions).
Notification Preview

Preview Records

<table>
<thead>
<tr>
<th>Event Creator</th>
<th>Preview Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Administrator</td>
<td>CHG00000001</td>
</tr>
</tbody>
</table>

Users

ITIL User

Subject

Change Request CHG00000001 notification -- Rollback Oracle Version

Body

Short description: Rollback Oracle Version
Click here to view Change Request: CHG00000001

State: New
Category: Software
Configuration item: Sales Force Automation
Opened by: System Administrator
Assignment group:
Assigned to: ITIL User

Description:
Performance of the Siebel SFA software has been severely degraded since the upgrade performed this weekend.

We moved to an unsupported Oracle DB version. Need to rollback the Oracle Instance to a supported version.
Comments and Work notes:
### Table 706: Notification preview

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Creator</td>
<td>The user triggering the notification for the purpose of the preview. The event creator defaults to the user who clicked <strong>Preview Notification</strong>. You can change the creator as needed. You can change the preview record as needed to see the changes in the notification content.</td>
</tr>
<tr>
<td>Preview Record</td>
<td>The record triggering the notification for the purpose of the preview. The preview record defaults to one of the records in the table specified in the <strong>Table</strong> field on the Email Notification form. You can change the preview record as needed to see the changes in the notification content.</td>
</tr>
</tbody>
</table>
| Event type          | The type of event that triggers the notification. This choice list appears if you preview an event-triggered notification. Select one of the following:  
  - **Generated Event**: Preview the notification with a generic event that the previewer creates. This does not actually generate an event record.  
  - **Existing Event**: Preview with an existing event record in the instance. If you select this option, select the event in the **Event Record** field. |
<p>| Event Record        | An existing event to preview an event-created notification. This option appears if you select <strong>Existing Event</strong> as the event type (for event-triggered notifications only). |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Users               | The users who will receive the notification, as specified in the **Who will receive** section of the Email Notification form:  
  - All users that you specify on the form appear, but only the users that will actually receive the notification with the current preview settings appear in black text.  
  - Users that are specified but for whatever reason will not receive the notification appear in red, strikethrough text. Place the cursor over any of these names to see the reason the user will not receive the notification. For example, one reason could be that the user's notification settings are disabled. |
| Subject and Body    | The content of the notification as defined by the template. The **Subject** and **Body** sections on the preview display the content in the corresponding **Subject** and **Message** fields on the template.  
  If the template includes a link to the record that triggered the notification, the **Preview Record** link is used. Click the link to go to that record.                                                                                           |

5. Click the X at the top of the preview window to close it.

6. Make the necessary changes to the notification or template, if necessary.

**Time zone for email notifications**

A system property controls the time zone that the instance uses for the date and time stamp of a message.

The date and time stamp of a notification uses the system time zone, not the time zone of any recipient. The email property *glide.email.append.timezone* controls whether to append the time zone. If true, the system time zone of the instance is appended to any dates or date/times in outbound email messages (for example, 2010-07-02 04:01:14 PST).

**Specify alternative outbound email addresses for notifications**

By default, the system sends all outbound email notifications from the default email address of the instance, but you can specify an alternative address.

Role required: admin

For organizations that need to send email messages from specific email addresses, such as from multiple service desks, or they want to send notifications in different languages, the platform supports configuring multiple outbound addresses.

1. Navigate to **System Notification > Email > Notifications**.
2. Select an existing notification record for the desired event, such as **Incident Closed**.
3. Create a copy of this notification for each outbound email address.
4. Open one of the notification copies, and click the **Advanced view** related link.
5. In the **What it will contain** section, add an email address to the From field that is different from the default instance address.

6. Add a different email address than the **From** address to the **Reply to** field if you want replies to this notification to go to a different address.

   The system checks the **From** field for an address. If this field is empty, then the system uses the default address for the instance. If the **Reply to** field is empty, then all replies are sent to the address from which the notification was sent. If the **Reply to** field contains an email address, then the system sends all replies to the notification to this address.

7. Create mutually-exclusive conditions for notifications of the same type, so only the desired notification is sent when the event is fired.

   For example, if the **Company** is a certain value, then the notification comes from a unique email address entered in the **From** field.

8. Click **Update**.

---

**Specify an outbound email address for a particular language**

You can specify a different email address for each language your instance supports.

Role required: admin

1. Create or copy a notification record for the desired event.

2. In the **What will it contain** section, enter a new email address in the **From** field.

3. Create the **Subject** and **Message** content in the desired language.

4. In the **When to send** section, create a condition as follows:

   1. In the list of **Condition** fields, select **Show Related Fields** from the bottom of the choice list.

   2. From the choice list of **Related Fields**, select the field that identifies the recipient.

      For example, select **Caller > User** fields to send the notification to the user who called in an incident, or **Assigned to > User** fields to send the notification to the user to whom an incident is assigned.

   3. From the choice list of user fields, select **Language**.

   4. Select the **is** operator.

   5. Complete the condition by selecting the language of the desired user.

5. Click **Update**.

   All notifications for that event originate from the specified email address and go out in the language of the recipient.

---

**Scripting for email notifications**

Email scripts allow for business rule-like scripting within an outbound email message.

With mail scripts, you can dynamically change the email output of your system based on different criteria. Mail scripts allow you to perform simple tasks, such as displaying incident data, and complex ones, such as making advanced database queries.

You can add a **${mail_script:script name}** embedded script tag to the body of the email notification or template, replacing script name with the name of the script you created. This makes it easy to use the same scripts in multiple email notifications or templates.

If you manually enter a mail script bounded by **<mail_script>** and **</mail_script>** in the body of a new or converted email notification or template, and then attempt to save the record, a message asks whether the mail script should be converted. In many cases, an unconverted mail script fails to run from inside the HTML editor. If you select **Yes**, the script is added to the Email
Script [sys_script_email] table and is automatically replaced in the body with an embedded script tag (${mail_script:script_name}).

**JavaScript in emails**

Create mail scripts in **System Notifications > Email > Notification Email Script**, and refer to them by using ${mail_script:script_name} in the script field.

To print text into the body of the message, use the template.print("a string") function.

```javascript
var gr = new GlideRecord("sc_req_item");
gr.addQuery("request", current.sys_id);
gr.query();
while(gr.next()) {
  template.print(gr.number + " " + gr.cat_item.getDisplayValue() + ", Stage: " +
  gr.stage.getDisplayValue() + "<br />");
}
```

**Figure 766: JavaScript in templates**

The event.parm1 and event.parm2 parameters that come from the originating event can also be used within the <mail_script>. To use these, enter event.parm1 or event.parm2 in the mail script.

**Links to records in email notifications**

Adding the ${URI} parameter to an outbound email body or template creates a link to a specific record.

When a user clicks the word LINK, the instance prompts the user to log in if not already logged in, and then redirects the user to the record specified in the URI.

**Figure 767: Link displayed by $URI parameter**

The $URI parameter has an extension called the $URI+ format to specify additional arguments in the email link, such as sysparm terms, in addition to the automatically created URI. For example (whitespace added for improved readability):

```javascript
${URI+&sysparm_scriptlet=current.assigned_to=gs.getUserID()&sysparm_scriptlet_condition=current.assigned_to.nil()&sysparm_view=incident_active}
```

This example executes the JavaScript:

```javascript
current.assigned_to=gs.getUserID()
```
when the condition of

```javascript
current.assigned_to.nil()
```

is satisfied. Additionally, the script sets the view to `incident_active`.

**Enable links to records**

Adding the special `${URI}` parameter to an outbound email body or template creates a link to a specific record.

When a user clicks on the word **LINK**, the instance prompts the user to log in if not already logged in, and then redirects the user to the record specified in the URI.

**Figure 768: URI email notification**

The `${URI}` parameter has an extension called the `${URI+}` format to specify additional arguments in the email link, such as `sysparm` terms, in addition to the automatically created URI. For example (whitespace added for improved readability):

```javascript
$(URI+&sysparm_scriptlet=current.assigned_to=gs.getUserID()
&sysparm_scriptlet_condition=current.assigned_to.nil()
&sysparm_view=incident_active)
```

This example executes the JavaScript:

```javascript
current.assigned_to=gs.getUserID()
```

when the condition of

```javascript
current.assigned_to.nil()
```

is satisfied. Additionally, the script sets the view to `incident_active`.

**Change the link text**

To show the display value of the record as the link text instead of the word **LINK**, use the `${URI_REF}` parameter instead of the `${URI}` parameter.

**Figure 769: URI_REF email notification**
For example, if the URL displays an incident record, the link text is the incident number, which is the display value for incidents. If the URL displays a user record, then the link text is the user name.

**Link to related records**

A notification can link to a related record by specifying a reference field in front of the `${URI}` or `${URI_REF}` parameters.

Format the related record link as follows:

- `${<reference field that contains the related record you want to display>.URI}`
- `${<reference field that contains the related record you want to display>.URI_REF}`

For example:

<table>
<thead>
<tr>
<th>Related record to provide link to</th>
<th>Notification record table</th>
<th>Reference field</th>
<th>Samples</th>
</tr>
</thead>
</table>
| Related task record to be approved from an approval notification | Approval [sysapproval_approver] | Approval for [sysapproval] | • ${sysapproval.URI}  
  • ${sysapproval.URI_REF} |
| Related problem record in an incident notification | Incident | Problem [problem_id] | • ${problem_id.URI}  
  • ${problem_id.URI_REF} |

For example, the following notification template produces the email links in the picture below:

```plaintext
Click here to view Incident: ${URI_REF}
Click here to view Related Problem: ${problem_id.URI_REF}
```

**Figure 770: Related record link**

**Content page links in email notifications**

Links to CMS pages can be put in notifications to make it easy for the reader to access the pages.

The link takes the following format: `${CMS_URI+<site>/<page>}`.

For example, to link the email recipient to a page called Incident in the content site ESS, with the current incident as the target document, use the following format: `${CMS_URI+ess/incident_detail}`

The resulting email URL has this format:

```
https://<instance name>.service-now.com/ess/incident_detail.do?
sysparm_document_key=incident,46e18c0fa9fe19810066a0083f76bd56
```

**Link to content pages**

You can create links to content management pages using the following format: `${CMS_URI+<site>/<page>}`.

For example, to link the email recipient to a page called Incident in the content site ESS, with the current incident as the target document, use the following format:
The resulting email URL has this format:

https://<instance name>.service-now.com/ess/incident_detail.do?sysparm_document_key=incident,46e18c0fa9fe19810066a0083f76bd56

Example scripting for email notifications

Examples of scripting for email notifications.

**Scripting examples for email notifications**

A simple text string is the most basic example of the way a mail script works. This script prints out "Incident number - INC00001".

```
template.print("Incident number - "+ current.number);
```

More advanced scripts, like this one, can be found by browsing through the base system email templates.

```
template.print("Summary of Requested items:<br />
var gr = new GlideRecord("sc_req_item");
gr.addQuery("request", current.sysapproval);
gr.query();
while(gr.next()) {
    template.print(gr.number + ": " + gr.quantity + " X " +
    gr.cat_item.getDisplayValue()
    + " at " + gr.cat_item.price.getDisplayValue() + " each
<br />
}
```

To dynamically change field values within an email, use the following functions within `<mail_script>` syntax:

```
... email.setFrom(current.caller_id.email);
email.setReplyTo("joe.employee@yourcompany.com");
email.setSubject("This is the new subject line");
email.setBody("This is the new body");
...
```

Using the instance_name property ensures that the notification still works when migrated between instances.

```
dothis();
function dothis(){
    var gr = new GlideRecord('sys_attachment');
gr.addQuery('table_sys_id',current.sys_id);
gr.query();
while(gr.next()){
    template.print('Attachment: <a href="https://
gs.getProperty('instance_name')+'.service-now.com/sys_attachment.do?sys_id='+ gr.sys_id+'">'+
gr.file_name+'</a>');}
```

You can specify copied and blind copied recipients by using the email object within a mail script.

```
//email.addAddress(type, address, displayname);
email.addAddress("cc","john.copy@example.com","John Roberts");
email.addAddress("bcc","john.secret@example.com","John Roberts");
```
The following is an example script to add users from watch_list as copied recipients.

```java
if(!current.watch_list.nil()){
    //get watch list addresses and add to cc
    var watcherIds = current.watch_list.split(",");

    //get user records
    var user = new GlideRecord("sys_user");
    user.addQuery("sys_id", watcherIds);
    user.addQuery("notification",2);
    //email
    user.addQuery("email","!=");  //!
    user.query();

    while(user.next()){
        //add to cc list
        email.addAddress("cc", user.email, user.getDisplayValue());
    }
}
```

Mail script API
Certain variables are available when processing mail_script scripts.

### Table 708: Mail script variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Object Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>template</td>
<td>Handles printing from the mail script to the email message.</td>
</tr>
<tr>
<td></td>
<td>template.print(&quot;message&quot;); //outputs message to the email body.</td>
</tr>
<tr>
<td></td>
<td>template.space(&quot;number of spaces&quot;); //outputs spaces to the email body.</td>
</tr>
<tr>
<td>email_action</td>
<td>GlideRecord object for the email notification (sysevent_email_action).</td>
</tr>
<tr>
<td>event</td>
<td>GlideRecord object for the event that fired the notification (sysevent).</td>
</tr>
<tr>
<td>Variable</td>
<td>Object Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------</td>
</tr>
<tr>
<td>email</td>
<td>EmailOutbound object</td>
</tr>
</tbody>
</table>

Available methods:

- `addAddress(String type, String address, String displayname):` type can be cc or bcc.
- `setFrom(String address):` override the sender address.
- `setReplyTo(String address):` override the reply to address.
- `setSubject(String subject):` override the subject of the message.
- `setBody(String message):` override the body of the message.

The email address that is passed by setFrom and setReplyTo needs to be in a valid form such as helpdesk@sn.com or Display Name <helpdesk@sn.com>. If the email address includes a 'Display Name', then that value overrides the instance's display name.

Useful attachment scripts

This is a searchable version of the Useful Attachment Scripts.
For an easy-to-navigate version, visit the Useful Scripts portal.

**Caution:** The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.
## Table 709: Useful attachment scripts

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Table</th>
<th>Description</th>
<th>Parameters</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Attachments from Record to Record</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GlideSysAttachment.copy('sourcetable','sys_id','destinationtable','sys_id');</td>
</tr>
<tr>
<td>Delete Duplicate Attachments</td>
<td>business rule, scheduled job, background script</td>
<td>sys_attachment</td>
<td>This script will delete duplicate images located in the attachments table.</td>
<td>None</td>
<td>function fixDuplicateImages(){var gr =new GlideRecord('sys_attachment'); gr.addQuery('table_name','LIKE','ZZ_YY%'); gr.orderBy('table_sys_id'); gr.orderByDesc('sys_created_on'); gr.query();var lastID = 'not_a_match';var lastFile = 'not_a_match';while(gr.next()){var isDuplicate = (lastID == gr.table_sys_id) &amp;&amp; (lastFile == gr.file_name); lastID = gr.table_sys_id; lastFile = gr.file_name; gs.print(gr.table_sys_id + ' ' + gr.table_name + ' ' + gr.file_name + ' ' + gr.sys_created_on + ' ' + isDuplicate);if(isDuplicate) gr.deleteRecord();}</td>
</tr>
<tr>
<td>Display Whether Tasks Have Attachments in List View</td>
<td>Business Rule</td>
<td>Attachments [sys_attachment]</td>
<td>Displays whether tasks have attachments when viewed in the record list view.</td>
<td></td>
<td>function checkAttachment(){if(current.operation()=='insert') {hasAttachment('true');} // if deleting attachment check for other attachments if(current.operation()=='delete') {var timeNow3 =new GlideDateTime(); gs.log('has_attachment br: gliderecord query start date time is: '+ timeNow3.getNumericValue(),'jwtest');var attachCount =new GlideAggregate('sys_attachment'); attachCount.addQuery('table_sys_id',current.sys_id); attachCount.addAggregate('COUNT'); attachCount.query(); var numAttachments = '0'; // if no other attachments task does not have attachment if(attachCount.next()) {numAttachments = attachCount.getAggregate(&quot;COUNT&quot;);if(numAttachments&gt;0){hasAttachment='true';}}else{hasAttachment='false';}var timeNow4 =new GlideDateTime(); gs.log('has_attachment br: gliderecord query start date time is: '+ timeNow4.getNumericValue(),'jwtest');} function hasAttachment(answer){var task =new GlideRecord('task'); task.addQuery('sys_id',current.table_sys_id); task.query(); if(task.next()) {task.u_has_attachment = answer; task.autoSysFields(false); // Don't set the lastUpdatedTime or the Simultaneous Update Alert will likely get triggered task.setWorkflow(false); // Don't allow other business rules to run, otherwise multiple notifications will likely be sent task.update();}}}</td>
</tr>
</tbody>
</table>
Attachment Logging

Whenever a user downloads an attachment, the action writes an attachment.read event record to the event log. If desired, you can process these events with a Script Action or an Email Notification. This can be useful if you want to do something when an attachment is read. For example, you can record when and by whom certain attachments are downloaded. For this functionality, the current variable must point to a sys_attachment record, and the event record must use the following parameters:

- parm1: File name
- parm2: Table name

Notification examples
There are several examples that illustrate how to set up email and SMS notifications.

The two processes are identical, with the exception of the length and content of the message. SMS messages do not permit user response to links and are limited to 140 characters. Use the SMS alternate field in the Email Template and Email Notification forms to create a brief text message for SMS devices.

Baseline email notifications
The baseline system provides several email notifications.

Some events listed in this table do not appear in a business rule and are fired by other conditions in the platform. Some events are hardcoded and are not user configurable.

Table 710: Baseline email notification descriptions

<table>
<thead>
<tr>
<th>Email notification</th>
<th>Description</th>
<th>Triggering event</th>
<th>Business Rule Controlling Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appointment Invite</td>
<td>Type: Meeting Invitation</td>
<td>itil_appointment.inserted</td>
<td>Global business rule</td>
</tr>
<tr>
<td></td>
<td>Adds a meeting invitation to the recipient’s calendar by sending an iCalendar formatted email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Appointment Update</td>
<td>Type: Meeting Invitation</td>
<td>itil_appointment.updated</td>
<td>Global business rule</td>
</tr>
<tr>
<td></td>
<td>Updates an existing meeting in the recipient’s calendar by sending an iCalendar formatted email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approval Rejected</td>
<td>Type: EMAIL</td>
<td>approval.rejected</td>
<td>approver events</td>
</tr>
<tr>
<td></td>
<td>An approval has been rejected – includes the approver’s name.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approval Rejected by Other</td>
<td>Type: EMAIL</td>
<td>approval.rejected.by.othe approval events</td>
<td>An approval has been rejected – includes the approver’s name.</td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
<td>Triggering event</td>
<td>Business Rule Controlling Event</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Approval Request</td>
<td>Type: EMAIL</td>
<td>approval.inserted</td>
<td>approver changes</td>
</tr>
<tr>
<td></td>
<td>Sends an email for the recipient to reply with an approval decision.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the approval email, the recipient selects a link that builds the appropriate reply email.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Approval Rejected</td>
<td>Type: EMAIL</td>
<td>request.approval.rejected</td>
<td>approval events</td>
</tr>
<tr>
<td></td>
<td>A catalog request has been rejected – includes the approver’s name.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Approval Request</td>
<td>Type: EMAIL</td>
<td>request.approval.cancelled</td>
<td>approver changes</td>
</tr>
<tr>
<td></td>
<td>A catalog request for which you were an approver has been cancelled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Approval Request</td>
<td>Type: EMAIL</td>
<td>request.approval.inserted</td>
<td>approval events</td>
</tr>
<tr>
<td></td>
<td>A catalog request for which you are an approver has been made.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certificate Expired</td>
<td>Type: EMAIL</td>
<td>certificate.expired</td>
<td>certificate events</td>
</tr>
<tr>
<td></td>
<td>Notification that the X.509 certificate has expired.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certificate Expiring</td>
<td>Type: EMAIL</td>
<td>certificate.expiring</td>
<td>certificate events</td>
</tr>
<tr>
<td></td>
<td>Notification that the X.509 certificate is expiring in N days.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change approved</td>
<td>Type: EMAIL</td>
<td>change.approved</td>
<td>Change events and task events</td>
</tr>
<tr>
<td></td>
<td>A change request has been approved.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change assigned to me</td>
<td>Type: EMAIL</td>
<td>change.assigned</td>
<td>Change events</td>
</tr>
<tr>
<td></td>
<td>A change request has been assigned to you.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
<td>Triggering event</td>
<td>Business Rule Controlling Event</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Change assigned to my group</td>
<td>Type: EMAIL</td>
<td>change_assigned.to.group</td>
<td>Change events</td>
</tr>
<tr>
<td></td>
<td>A change request has been assigned to your group.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change commented (to assignee)</td>
<td>Type: EMAIL</td>
<td>change_commented</td>
<td>Change events</td>
</tr>
<tr>
<td></td>
<td>A comment has been added to a change request. The person assigned to the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>change request receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change commented (unassigned)</td>
<td>Type: EMAIL</td>
<td>change_commented</td>
<td>Change events</td>
</tr>
<tr>
<td></td>
<td>A comment has been added to a change request. The assignment group</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>assigned to the change request receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Notification</td>
<td>Type: EMAIL</td>
<td>label.notify</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notification of a change in the fields label in a form.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change rejected</td>
<td>Type: EMAIL</td>
<td>change_rejected</td>
<td>Change events and task events</td>
</tr>
<tr>
<td></td>
<td>A change request has been rejected.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Task worknoted (unassigned)</td>
<td>Type: EMAIL</td>
<td>change_task_worknoted</td>
<td>Change task events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a change task. The assignment group assigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to the change task receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Task worknoted (to assignee)</td>
<td>Type: EMAIL</td>
<td>change_task_worknoted</td>
<td>Change task events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a change task. The person assigned to the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>change task receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
<td>Triggering event</td>
<td>Business Rule Controlling Event</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Change Task assigned to my group</td>
<td>Type: EMAIL</td>
<td>change_task.assigned.to.group</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A change task has been assigned to your group.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Task assigned to me</td>
<td>Type: EMAIL</td>
<td>change_task.assigned</td>
<td>Change task events</td>
</tr>
<tr>
<td></td>
<td>A change task has been assigned to you.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change worknoted (to assignee)</td>
<td>Type: EMAIL</td>
<td>change.worknoted</td>
<td>Change events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a change request. The person assigned to the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>change request receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change worknoted (unassigned)</td>
<td>Type: EMAIL</td>
<td>change.worknoted</td>
<td>Change events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a change request. The assignment group assigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to the change request receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email assigned to</td>
<td>Type: EMAIL</td>
<td>incident.assigned</td>
<td>incident.events</td>
</tr>
<tr>
<td></td>
<td>An incident has been assigned to you (the recipient of the email).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
<td>Triggering event</td>
<td>Business Rule Controlling Event</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>------------------</td>
<td>---------------------------------</td>
</tr>
</tbody>
</table>
| Email assigned to (sc_task) | Type: EMAIL  
A task has been assigned to you (the recipient of the email). | sc_task.assigned.to.user sc_task_events | |
| Email assigned To Group | Type: EMAIL  
An incident has been assigned to an assignment group of which you are a member (the recipient of the email). | incident.assigned.to.group incident_events | |
| Email assigned to group (sc_task) | Type: EMAIL  
A Service Catalog task has been assigned to an assignment group of which you are a member (the recipient of the email). | sc_task.assigned.to.group sc_task_events | |
| Incident Closed | Type: EMAIL  
An incident opened by you (the recipient of the email), has been closed. | incident.updated incident_events | |
<table>
<thead>
<tr>
<th>Email notification</th>
<th>Description</th>
<th>Triggering event</th>
<th>Business Rule Controlling Event</th>
</tr>
</thead>
</table>
| Incident Commented | Type: EMAIL  
An incident opened by you (the recipient of the email) has had comments added. This notification uses a template for an employee self-service (ESS) user. | incident.commented | incident events |
| Incident Commented | Type: EMAIL  
An incident assigned to you (the recipient of the email) has had comments added. This notification uses a template for an ITIL user. | incident.commented | incident events |
| Incident Opened    | Type: EMAIL  
An incident has been opened for you (the recipient of the email) by someone else. This notification uses a template for an employee self-service (ESS) user. | incident.inserted | incident events |
| Incident Opened &  Unassigned | Type: EMAIL  
An incident has been opened and is unassigned. This notification uses a template for an ITIL user. | incident.inserted | incident events |
| Incident Resolved  | Type: EMAIL  
An incident opened by you has been resolved, and feedback is required to determine if the incident should be closed. | incident.updated | incident events |
<table>
<thead>
<tr>
<th>Email notification</th>
<th>Description</th>
<th>Triggering event</th>
<th>Business Rule Controlling Event</th>
</tr>
</thead>
</table>
| Knowledge Closed Created    | Type: EMAIL  
A contributor's submission to the Knowledge Base was accepted and an article was created. | kb.submission.closed_created_event |                  |
| Knowledge Closed Duplicate  | Type: EMAIL  
A contributor’s submission to the Knowledge Base was determined to be a duplicate, and no article was created. | kb.submission.closed_duplicate_event |                  |
| Knowledge Closed Invalid    | Type: EMAIL  
A contributor's submission to the Knowledge Base was determined to be invalid (unusable). | kb.submission.closed_invalid_event |                  |
| Notify Change Calendar      | Type: Meeting Invitation  
Notifies the recipients of the schedule for a change request and exports the schedule to the Microsoft Outlook calendar. The email is in the format of iCalendar formatted email. | change.calendar.notify | change events |
| Notify Change Calendar Remove| Type: Meeting Invitation  
Notifies the recipients that a scheduled change has been closed or assigned to someone else and removes the entry from the Microsoft Outlook calendar. The email is in the format of iCalendar formatted email. | change.calendar.notify.remove | change events |
<table>
<thead>
<tr>
<th>Email notification</th>
<th>Description</th>
<th>Triggering event</th>
<th>Business Rule Controlling Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Task assigned to me</td>
<td>Type: EMAIL</td>
<td>problem_task.assigned</td>
<td>Problem task events</td>
</tr>
<tr>
<td></td>
<td>A problem task has been assigned to you.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem Task assigned to my group</td>
<td>Type: EMAIL</td>
<td>problem_task.assigned.to.group</td>
<td>Problem task events</td>
</tr>
<tr>
<td></td>
<td>A problem task has been assigned to your group.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem Task worknoted (to assignee)</td>
<td>Type: EMAIL</td>
<td>problem_task.worknoted</td>
<td>Problem task events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a problem task. The person assigned to the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>problem task receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem Task worknoted (unassigned)</td>
<td>Type: EMAIL</td>
<td>problem_task.worknoted</td>
<td>Problem task events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a problem task. The assignment group assigned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to the problem task receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem worknoted (to assignee)</td>
<td>Type: EMAIL</td>
<td>problem.worknoted</td>
<td>Problem events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a problem. The person assigned to the problem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem worknoted (unassigned)</td>
<td>Type: EMAIL</td>
<td>problem.worknoted</td>
<td>Problem events</td>
</tr>
<tr>
<td></td>
<td>A work note has been added to a problem. The assignment group assigned to</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the problem receives an email notification.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reminder Insert</td>
<td>Type: Meeting Invitation</td>
<td>reminder.notify</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a calendar reminder regarding an open task. The email is in the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>format of iCalendar formatted email.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
<td>Triggering event</td>
<td>Business Rule Controlling Event</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Reminder Insert</td>
<td>Type: Meeting Invitation&lt;br&gt;A task has been closed or deleted. This notification removes the task reminder from Outlook. The email is in the format of iCalendar formatted email.</td>
<td>reminder.notify.delete</td>
<td></td>
</tr>
<tr>
<td>Reminder Insert Email</td>
<td>Type: EMAIL&lt;br&gt;Sends the recipient a reminder email about a specific task.</td>
<td>reminder.notify.email</td>
<td></td>
</tr>
<tr>
<td>Request Approved</td>
<td>Type: EMAIL&lt;br&gt;A Service Catalog request, opened by the recipient, has been approved.</td>
<td>sc_request.approved</td>
<td>sc_request events</td>
</tr>
<tr>
<td>Request Assigned</td>
<td>Type: EMAIL&lt;br&gt;A Service Catalog request has been assigned to the recipient.</td>
<td>sc_request.assigned</td>
<td>sc_request events</td>
</tr>
<tr>
<td>Request Completed</td>
<td>Type: EMAIL&lt;br&gt;A Service Catalog request, opened by the recipient, has been completed.</td>
<td>sc_request.updated</td>
<td>sc_request events</td>
</tr>
<tr>
<td>Request Item Assigned</td>
<td>Type: EMAIL&lt;br&gt;An item requested from the Service Catalog has been assigned to you.</td>
<td>sc_req_item.assigned</td>
<td>sc_request events</td>
</tr>
<tr>
<td>Request Item Delivery</td>
<td>Type: EMAIL&lt;br&gt;An item requested from the Service Catalog by the recipient is being delivered.</td>
<td>sc_req_item.delivery</td>
<td>sc_request events</td>
</tr>
<tr>
<td>Email notification</td>
<td>Description</td>
<td>Triggering event</td>
<td>Business Rule Controlling Event</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Request Opened on Behalf</td>
<td>Type: EMAIL</td>
<td>sc_request.requested_for</td>
<td>sc_request_events</td>
</tr>
<tr>
<td></td>
<td>A Service Catalog request has been opened on behalf of the recipient.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset Password</td>
<td>Type: EMAIL</td>
<td>reset.password</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The recipient’s password has been reset as requested.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduled Import Completed</td>
<td>Type: EMAIL</td>
<td>scheduled_import_set.completed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A scheduled import set has completed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Upgraded</td>
<td>Type: EMAIL</td>
<td>system.upgraded</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The recipient’s system has been upgraded.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task approved</td>
<td>Type: EMAIL</td>
<td>task.approved</td>
<td>Change events and task events</td>
</tr>
<tr>
<td></td>
<td>An ITIL task has been approved.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text Index Completed</td>
<td>Type: EMAIL</td>
<td>text_index.complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A scheduled system index has completed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unscheduled Change</td>
<td>Type: EMAIL</td>
<td>cmdb.unscheduled.change</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A named configuration item has changed, and no active change request exists.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notification example: simple reminder

In this notification example, set up a very simple reminder that sends a user an email at a specified time, based on information in an incident.

Role required: admin

We will create the following:

- A custom date/time field on the Incident form
- A business rule to define an event based on the new field
- An email notification to react to that event

1. Create a custom Date/Time field on the Incident form labeled Reminder, with a field name of u_reminder
2. Define an event that is triggered when the **Reminder** field gets a value. We create an event called `incident.reminder` that is triggered at a specific time. We use the `gs.eventQueueScheduled()` function, which is similar to the standard event creation function, but can pass a parameter to specify when the event is triggered. In this case, the event is triggered at the date and time specified in the new **Reminder** field.

**Note:** Be sure to register your new event. If you do not, the system will not know to look for it.
3. Create an email notification record that reacts to the new incident.reminder event. This brief notification message is suitable for SMS as well.

```java
gs.eventQueueScheduled("incident.reminder", current, gs.getUserId()
    ( ), gs.getUserName( ), current.u_reminder);
```
Note: This is a very simple reminder. There is no condition in place to delete or reschedule the reminder event if the incident is deleted or if the date and time in the Reminder field is changed. The business rule, as presented, will schedule a new event every time the Reminder field is updated. Nothing is configured to display a reminder that is currently scheduled for an incident. However, this reminder is very simple and can be set up in just a few minutes.

Create a concept topic to introduce the background needed to perform this process or task.

Notification example: assignment notification

In this notification example, enhance the To Do application to send an email notification when someone is assigned a task.

Role required: admin

The email notification is triggered by an event, which is triggered by a business rule. Events can be triggered by business rules and, in turn, reacted to elsewhere. This event is triggered by the business rule created in the next procedure and will, in turn, trigger the email notification.

1. Navigate to System Policy > Events > Registry.
2. Click the New to create an event registry record.
3. Select the **To_Do** table.

4. Type **A To Do has been assigned** in the **Description** field.
   
   **Fired by** is basically a comment to remind you where the events come from. We select **Business Rule**.

5. Click **Submit**.

6. Navigate to **System Definition > Business Rules**.

7. Click **New** to create a business rule.

8. Fill in the fields as follows:
   
   - **Name**: To Do Assigned
   - **Table**: to_do
   - **Run at**: server
   - **When**: after
   - **Insert**: true
   - **Update**: true

9. Enter a condition to trigger the business rule whenever the **Assigned to** field is changed or initially populated:
   
   ```
   current.u_assigned_to.changes()
   ```

10. Enter a script to add a to_do.assigned event.

11. In addition to the event name and the current record, specify two additional parameters (user’s ID and Name).

    ```
    gs.eventQueue("to_do.assigned", current, gs.getUserID(), gs.getUserName());
    ```
12. Click Submit.
13. Test that the business rule is creating the event as follows:
   1. Change the value in the Assigned to field of an existing To Do record or create a new record and specify an Assigned to.
   2. Navigate to System Policy > Events > Event log and look for an event with the name to_do.assigned.
   3. To make this easier, you can sort the event log in descending order by created.

14. Navigate to System Notification > Email > Notifications.
15. Click New to create an email notification.
16. Fill in the following fields:
   - **Name**: To Do Assigned
   - **Event name**: to_do.assigned
   - **Table name**: to_do

   The **User field** specifies to whom the email should be sent. In this case, send it to the **Assigned to** user.
17. In **User field**, enter u_assigned_to.

   Sometimes, you do not want notifications to be sent to the person who triggered the notification. In this case, clear the **Send to event creator** check box. Select this option to see all the notifications.
18. Enter the **Subject** of the email as **To Do Assigned**.
19. Under **Select variables**, click **Due Date**.
You will see text added to the message text box which causes the value of the due date to be inserted. You can change the label (Due Date in this case) which was provided automatically. Use the same method to add two more variables to the message text: Priority and Short Description.

20. Click Submit.

21. **Warning:** Do not use your own POP server for this exercise. You should have a test account set up on your mail server (see the administrator of your mail system). It may download the contents of your POP account. You may choose to skip configuring Glide to use an email server and just check the Glide email log instead of sending an actual email (see final step below).

Navigate to **System Properties > Email**.

22. Under **SMTP Server Settings**, enter the outgoing mail server, outgoing mail address, the mail server password, and outgoing mail display name.

23. If you are using a POP server, enter the POP server, incoming POP mail account name (do not include the @ and server name here), and incoming POP mail account password.

At the bottom of the form is a space where a test email account can be provided to direct emails to it instead of the actual email addresses. Rather than test emails sent to the test users, place your own email address here.
24. Click **Save**.
25. Test the assignment notifications.
   1. Assign some **To Do** records.
   2. Navigate to **System Logs Email** to see that status of generated emails.

You should receive email notifications in a few moments.

Notification example: approval request

In this notification example, a change request approval is requested, which results in an update to the **Approval [sysapproval_approval]** table.

**Role required: admin**

The **approval events** Business Rule is executed which creates the **approval.inserted** event. The Approval Request email event is configured to process the **approval.inserted** event, which uses the change.itil.approve.role email template. This is part of an automatic response that enables the recipient to click a link in the email to send a pre-formatted response back to the system to either approve or reject the change request automatically.

1. Create an email notification that looks like this:
2. Create an email template like this:

The combination of the notification and template generates an email that looks something like the following:
Notice that the receiver of this email has the following links in the mail:

- A link to view the approval record
- A link to view all the details of the change request
- A link that generates an automatic email response to approve the change
- A link that generates an automatic email response to reject the change

Notification example: priority 1 incident update

In this notification example, configure the instance to notify specific users by email whenever an incident with a priority of 1 - Critical is updated, regardless of the changes that have been made.

Role required: admin

We will create an email notification to alert all recipients when the default system event incident.update is triggered for a priority 1 incident. The notification will display information of interest to the recipients, such as the incident number, category, assignees, and any comments that were added to the incident.

1. Navigate to System Notification Email Notifications, and then click New.
2. Configure the email notification record as follows.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Give your notification record a unique name, such as Incident of Priority 1 Updated.</td>
</tr>
<tr>
<td>User</td>
<td>Click the magnifying glass icon and select a recipient from the list of users. You can select only one user in this field.</td>
</tr>
<tr>
<td>Event name</td>
<td>Select the event to use for this notification. The incident.updated event is triggered by any change to an incident record and can be used to send our notification.</td>
</tr>
<tr>
<td>Field</td>
<td>Input Value</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User field</td>
<td>Enter the field name from the <strong>Incident</strong> table that defines the recipient for this notification. For example, you might send notification of an update to the person who opened the incident by typing <em>opened_by</em>.</td>
</tr>
<tr>
<td>Email template</td>
<td>Leave this field blank. We will cover the construction and use of templates in another exercise.</td>
</tr>
<tr>
<td>Group</td>
<td>Click the magnifying glass icon, and then select a group from the list of groups. You can select only one group in this field.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the appropriate database table, in this case <strong>Incident [incident]</strong>.</td>
</tr>
<tr>
<td>Group field</td>
<td>Enter the field name from the <strong>Incident</strong> table that defines the group to notify. For example, you might send notification of the update to the designated assignment group by typing <em>assignment_group</em>.</td>
</tr>
<tr>
<td>Weight</td>
<td>Weight determines the sending priority of each notification when more than one qualifies. This is an arbitrary scale that you set.</td>
</tr>
<tr>
<td>Send to event creator</td>
<td>Select this check box to send the notification to the person who performed the action that started the notification process if the person is also specified in the <strong>Users/groups in fields</strong>, <strong>Users</strong>, or <strong>Groups</strong> field. If the event creator is not specified in one of these fields, the event creator does not receive a notification regardless of the setting in this field. For new notifications, this option is selected by default.</td>
</tr>
<tr>
<td>Type</td>
<td>Select <strong>EMAIL</strong>.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box <em>(true)</em> to enable email notification.</td>
</tr>
<tr>
<td>Conditions</td>
<td><strong>Updated --&gt; is anything.</strong> This creates the condition under which the notification is sent. In the example, an update to any field in an incident record sends an email to all recipients.</td>
</tr>
<tr>
<td>Field</td>
<td>Input Value</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Subject</td>
<td>Type an appropriate subject line for your email. Select variables for your content from fields in the <strong>Incident</strong> table. Put the cursor in the subject line where you want the variable to appear, and then click the field name in the <strong>Select variables</strong> column. For this exercise, type the following subject line and insert the variable for the <strong>Priority</strong> field: An Incident of Priority ${priority} has been updated. In this notification, the variable ${priority} returns the value 1 – Critical.</td>
</tr>
</tbody>
</table>
| Message      | Construct a message that includes all the information you think the recipients need to know about the updated incident. Select the appropriate variables for your message from the fields in the **Incident** table. Put the cursor in the message where you want the variable to appear, and then click the field name in the **Select variables** column. The special character, `<hr/>`, creates a manual break in the message that sets off the Comments section. For this example, create the following message:  

<table>
<thead>
<tr>
<th><strong>Short Description:</strong></th>
<th>${short_description}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click here to view incident:</td>
<td>${URI}</td>
</tr>
<tr>
<td>Incident number:</td>
<td>${number}</td>
</tr>
<tr>
<td>Category:</td>
<td>${category}</td>
</tr>
<tr>
<td>Assigned to:</td>
<td>${assigned_to}</td>
</tr>
<tr>
<td>Assignment group:</td>
<td>${assignment_group}</td>
</tr>
<tr>
<td>&lt;hr/&gt;</td>
<td></td>
</tr>
<tr>
<td>Comments:</td>
<td>${comments}</td>
</tr>
<tr>
<td>List</td>
<td>Click the lock icon to open a list of recipients for the email notification. Click the magnifying glass icon and select names from the list of users. You can select as many users from the list as you want.</td>
</tr>
<tr>
<td>List field</td>
<td>Enter the field name from the <strong>Incident</strong> table that identifies a list that contains potential recipients. For example, to send the notification to users who are on the watch list for this issue, type watch_list.</td>
</tr>
</tbody>
</table>
### Field | Input Value
--- | ---
SMS alternate | Enter an abbreviated version of the email message that eliminates such things as the $(URI)$ link that requires recipient interaction and the $(comments)$ field. SMS messages can only display 140 characters.

Your email notification record should look like this:

3. **Note:** After you have created the email notification record, set up a test in your environment to make sure the intended recipients get the proper notification.

Create users in your platform who have email addresses you can monitor, and then create a group that includes one of these users.
4. Open a **Priority 1 – Critical incident**, and assign it to one of the users you created. Then select the group you created as the assignment group.

5. Open your email notification record and specify the recipients.
   1. Select one of the users you created from the lookup list in the **User** field.
   2. Type `assigned_to` for the **User field** value.
   3. Type `assignment_group` for the **Group field** value.

6. Update your Priority 1 incident by adding comments, and then click **Update**.

7. Check the email accounts of the user to whom you assigned the incident and the user member of the assignment group.

The email that is sent should have the same structure as the following sample:

---

### Email templates

Email templates enable administrators to create reusable content for the subject line and message body of email notifications.

Templates deliver consistent information on specific system activities and improve the efficiency of creating multiple email notifications for similar actions. If necessary, you can make minor changes on the fly to an email that uses a template by overriding the subject line and message body content in the notification form. A common practice is to define the message body in a template and create new subject lines for different types of recipients. There is no limit to the number of templates that you can create.

Email templates are created in rich HTML format, and administrators have the option of converting existing email templates to rich HTML (starting with the Eureka release). This format provides several advantages, including:

- Raw HTML content is converted into a WYSIWYG format.
- The content can be edited in a feature-rich HTML editor.
- Mail scripts are condensed into a single, easy-to-read line that can be reused in multiple email notifications.
- To prevent broken links, images linked using URLs relative to a particular instance are converted to absolute links.

**Create an email template**

You can create an email template with rich HTML formatting, rather than plain text.

**Role required: admin**

1. Navigate to **System Policy > Email > Templates**.
2. Click **New**.
3. Fill in the form fields (see table).
Figure 771: The Email Template form with the HTML editor

Geneva    ServiceNow    ServiceNow Platform

© 2017 ServiceNow. All rights reserved. 2807
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the template. For example, change.update.risk.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the name of the table involved. For example, Change Request [change_request].</td>
</tr>
<tr>
<td>Application</td>
<td>The type of scoped application.</td>
</tr>
</tbody>
</table>
| Subject          | Enter a subject line that explains the purpose of the email. Select the appropriate variables for the subject line from the fields available on the selected table. Place the cursor where you want the variable to appear, and click the field name in the **Select variables** column. For example: 

   `$(sys_class_name) $(number)
   with $(risk) risk has been
   assigned to you.`  

| Message HTML     | Enter the content of the email template message. You can use the HTML editor toolbar to format the HTML, and you can include variables from the **Select variables** column. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table such as an incident's short description or comments and work notes. **Notes:**

   - When a template is applied to a notification, the contents of this field are used if you selected **HTML and plain text** or **HTML** in the **Content Type** field in the Email Notification form.
   - You cannot use HTML code to control the appearance of the contents in the `$(comments)` variable.
   - If you want to include a link to the record that triggered the notification.  

<p>| Message Text     | Enter the notification message to send in plain text. <strong>Note:</strong> When a template is applied to a notification, the contents of this field are used if you selected <strong>HTML and plain text</strong> or <strong>Text</strong> in the <strong>Content Type</strong> field in the Email Notification form. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS alternate</td>
<td>Create a different message to be delivered to the recipient's SMS device. You might want a brief message, showing the most important information only. If this message field is blank, the contents of the <strong>Message Text</strong> field are used for the SMS message. <strong>Note:</strong> The message in this field is used when the message is sent to a device configured as SMS.</td>
</tr>
</tbody>
</table>

*Apply a template to an email notification*
After you create an email template, you can apply it to a notification.

Role required: admin

1. Navigate to **System Notification > Email > Notifications**.
2. Open the email notification record that should use the template.
3. Navigate to the **What it will contain** section.
4. From the **Email template** field, select the appropriate template.
5. Click **Update**.

*Convert an email template to rich HTML*
If you have an email template created prior to the Eureka release, you can convert it to rich HTML.

Role required: admin

1. Navigate to **System Policy > Email > Templates**.
2. Open the email template you want to convert.
Thank you for submitting $\{number\}. We appreciate your willingness to help out the knowledge management team.

Your article led us to generate the following new knowledge base articles:

$\{mail_script:kb_submission_script_1\}

The knowledge engineer had the following comments upon closure:

$\{close_notes\}
3. Click **Switch to Rich HTML Editor**.

When content is converted, these changes are made to the content:

- **HTML Editor:** Any raw HTML in the **Message** field is rendered as WYSIWYG text in the **Message HTML** field. The content can be edited in a feature-rich HTML editor.

- **Simplified Mail Scripts:** Any mail scripts in the body are automatically saved to the Email Script [sys_script_email] table and are replaced in the notification body with an embedded script tag. This makes the template body easier to read and makes it easier to reuse scripts in multiple email templates.
Thank you for submitting $\text{number}$. We appreciate your willingness to help out the knowledge management team. Your article led us to generate the following new knowledge base articles:
$\text{mail_script:kb_submission_script_1}$
The knowledge engineer had the following comments upon closure:
$\text{close_notes}$
4. Modify the message text, as needed, using the WYSIWYG editor or by editing the underlying HTML.

**Note:** The letter "P" at the bottom of the screen shows the location of your cursor within the Message field. In this case, the cursor is in a line containing an <HTML> tag.

**Note:** Email templates that are already formatted with rich HTML do not show the Switch to Rich HTML Editor button.

Whether you are working with templates converted from earlier versions or creating new templates in the rich HTML format, it is a good practice to write mail scripts in System Notification > Email > Notification Email Scripts. When the scripts are completed, a \$mail_script:script name\ embedded script tag should be added to the email template body. This makes it easy to use the same scripts in multiple email templates. All that needs to be copied and pasted from one template to the next is the embedded script tag.

If you manually enter a mail script in the body of a new or converted email notification or template, and then attempt to save the record, a message asks whether the mail script should be converted. Unconverted mail scripts often fail to run from inside the HTML editor. If you select Yes, the script is added to the Email Script [sys_script_email] table and is automatically replaced in the body with an embedded script tag.

You can view the mail scripts in their original form by opening the email template and clicking the Show Notification Scripts related link.

**Override a template value**
You can customize an email notification to override a specific value without changing the basic information contained in the template.

Role required: admin

1. Navigate to System Policy > Email > Templates.
2. Open an email template.
3. Enter a new value for the field you want to override.
4. Click Update.

For example, you might change the subject line to read The risk level of \$sys_class_name\ \$number\ has been raised to \$risk\). This would not change the basic information about the change request contained in the template, but would make the email notification appropriate for the change manager who needs to know when risk levels are escalated above a certain threshold.

**Construct an email message with a template**
Email templates provide a list of the fields from database tables that are available for constructing an email message.

There is also a template that allows a mailto:automatic response link which allows the receiver of the email can to simply choose a link to send a pre-formatted response back to the instance. An example is the email template that is used for notification that an approval is required. The base email is shown below:
### Notification - Approval Request [Advanced view]

<table>
<thead>
<tr>
<th>Name</th>
<th>Approval Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Approval(sysapproval,_approve)</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Content type</td>
<td>HTML and plain text</td>
</tr>
<tr>
<td>Include attachments</td>
<td>Yes</td>
</tr>
<tr>
<td>Omit watermark</td>
<td>No</td>
</tr>
<tr>
<td>Push Message Only</td>
<td>No</td>
</tr>
<tr>
<td>Importance</td>
<td>-- None --</td>
</tr>
<tr>
<td>From</td>
<td></td>
</tr>
<tr>
<td>Reply to</td>
<td></td>
</tr>
<tr>
<td>Push Messages</td>
<td>Approval Request</td>
</tr>
<tr>
<td>Email template</td>
<td>change审批流程</td>
</tr>
<tr>
<td>Subject</td>
<td></td>
</tr>
<tr>
<td>Message HTML</td>
<td>Select variables:</td>
</tr>
</tbody>
</table>
Notice the selection list on the right that shows the available fields from the sysapproval_approver table. All email templates get used based on an event that was created. Generally events are created based on Business Rules that execute when a database record is modified in some way. The database record being modified when the event is created is the record that is available for generating an email request.

In this case, a change request approval was being requested which results in an update to the sysapproval_approval table. The "approval events" Business Rule was executed which created the "approval. inserted" event. The "Approval Request" email event is defined to process the approval.inserted event and this email event is defined to use the change.itil.approve.role email template that is shown above. Notice that this template has two mailto: items specified. One for "mailto.approval" and one for "mailto.rejection." These are email templates that are used to construct an automatic response so that the email receiver can simply click a link in the email to automatically either approve or reject the change request.

Here is what the change.itil.approve.role template looks like using the rich HTML editor:
Figure 773: The change.itil.approve.role template

Name: change.itil.approve.role
Application: Global
Subject: $(sysapproval.sys_class_name) $(sysapproval) Approval Request

Message HTML:

Short Description: $(sysapproval.short_description)
Priority: $(sysapproval.priority)
Category: $(sysapproval.category)

$(mailto:sysapproval)

Click here to view Approval Request: $(URL)
Click here to view $(sysapproval.sys_class_name): $(sysapproval.URL)
And the `mailto.approval` template:

![Email Template](image)

**Figure 774: The mailto approval template**
Note: If you put text on the lines following the Click here to approve $\{sysapproval\}$ line, this forms the actual body of the email.

The above combination of templates would generate an email that looks something like the following:

Figure 775: Template generated email message

Notice that the receiver of this email has the following links in the mail:

• A link to view all the details of the change request
• A link to view the approval record
• A link that will generate an automatic email response to approve the change
• A link that will generate an automatic email response to reject the change

Add blank lines in an email template
You can easily add blank lines in an email template using HTML tags.

Judicious use of these HTML tags can make your email easier to read.

• You can use `<br/>` to insert a line break
• You can wrap paragraphs in `<p> . . . </p>` tags to format your email nicely.

Calendar integration
With email notifications, you can use import export maps to leverage information about records and integrate with Outlook or another calendar.

For example, an email notification can create a calendar event based on the planned start and end dates of a change request. To enable integration with a calendar, the following iCalendar variables are available to be added to an email template message and reference the email template from the notification. The variables must be added to the Message Text field.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>${dtstart}$</td>
<td>Start Date</td>
</tr>
<tr>
<td>${dtend}$</td>
<td>End Date</td>
</tr>
<tr>
<td>${location}$</td>
<td>Location</td>
</tr>
<tr>
<td>${alarm_time}$</td>
<td>Alert or reminder time</td>
</tr>
</tbody>
</table>
The instance computes the value of iCalendar variables using import and export maps for the following tables:

- change_request
- reminder
- itil_appointment
- incident

Figure 776: Import export maps

Each import export map can specify a different set of iCalendar fields. For example, the icalendar.change_request import export map only maps two iCalendar fields.

Figure 777: iCalendar change request mappings

The instance uses the **External Name** value as the variable name in the email template. For example, the icalendar.change_request import export map defines the **dtstart** and **dtend** variables.

<table>
<thead>
<tr>
<th>External name</th>
<th>Associated variable name</th>
<th>Table</th>
<th>Field mapped</th>
<th>Field label</th>
</tr>
</thead>
<tbody>
<tr>
<td>dtstart</td>
<td>${dtstart}</td>
<td>change_request</td>
<td>start_date</td>
<td>Planned start date</td>
</tr>
<tr>
<td>dtend</td>
<td>${dtend}</td>
<td>change_request</td>
<td>end_date</td>
<td>Planned end date</td>
</tr>
</tbody>
</table>

Map date fields to iCalendar variables
You can specify what fields provide the date information in calendar invitation notifications by changing the field mappings of the \texttt{dtstart} and \texttt{dtend} variables in the import export map for the iCalendar invitation.

Role required: admin

1. In the navigation filter, enter \texttt{sys_impex_map.list}.
2. Open the \texttt{icalendar.change_request} map or the map you want to edit.
3. In the Field Maps related list, click either the \texttt{end_date} or \texttt{start_date} mapped field to change the mapping for \texttt{dtstart} or \texttt{dtend}, as needed.
4. Change the Database field to the field you want to use to set the start date or end date.
5. Click Update.

Create iCalendar invitations for custom tables

To generate iCalendar invitations that use field values from custom tables, create an import export map that computes the values of the iCalendar fields.

Role required: admin

1. Create the custom table and fields using the Date/Time field type for the fields that map to the \$\texttt{dtstart} and \$\texttt{dtend} variables.
2. In the navigation filter, enter \texttt{sys_impex_map.list}.
3. Click New.
4. Set the following fields:
   - \textbf{Name}: Use the following naming convention: \texttt{icalendar.<table name>}. For example, \texttt{icalendar.u_my_custom_table}.
   - \textbf{Table}: Select the custom table you created.
   - \textbf{Type}: Select \texttt{icalendar}.
5. Right-click the form header and select Save.
6. In the Field Maps related list, click New.
7. In the Mapping Entry Wizard, select Mapping to a Database Field.
8. Create field mappings for \texttt{dtstart} and \texttt{dtend}. These variables are required.
   - For example, map the \textbf{External Name \texttt{dtstart}} to the \texttt{u.meeting_start_time} field in \texttt{u_my_custom_table}.
9. Click Submit.
10. Create field mappings for any other iCalendar fields as necessary.
11. Click Update.

Here are sample field mappings between iCalendar variables and custom fields in a custom table, \texttt{u_my_custom_table}:

<table>
<thead>
<tr>
<th>External name</th>
<th>Database field</th>
<th>Type</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{dtstart}</td>
<td>\texttt{u.meeting_start_time}</td>
<td>field</td>
<td>\texttt{icalendar.u_my_custom_table}</td>
</tr>
<tr>
<td>\texttt{dtend}</td>
<td>\texttt{u.meeting_end_time}</td>
<td>field</td>
<td>\texttt{icalendar.u_my_custom_table}</td>
</tr>
<tr>
<td>\texttt{description}</td>
<td>\texttt{u.meeting_description}</td>
<td>field</td>
<td>\texttt{icalendar.u_my_custom_table}</td>
</tr>
</tbody>
</table>

12. Create an email template that defines what to include in the iCalendar invitation.
   a) Set the Table field to the custom table you created.
b) In the **Message** text field, use the following format to define the iCalendar invitation.

```
BEGIN:VCALENDAR
PRODID:-//Service-now.com//Outlook 11.0 MIMEDIR//EN
VERSION:2.0
METHOD:REQUEST
BEGIN:VEVENT
ATTENDEE;ROLE=REQ-PARTICIPANT;RSVP=TRUE:MAILTO:${to}
DTSTART:${dtstart}
DTEND:${dtend}
UID:${sys_id}
DTSTAMP:${dtstamp}
DESCRIPTION:${description}
SUMMARY:${u_meeting_summary}
END:VEVENT
END:VCALENDAR
```

**Note:** Mail script is not allowed or processed in meeting invitation email templates.

<table>
<thead>
<tr>
<th>iCalendar template line</th>
<th>Required?</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN:VCALENDAR</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PRODID:-//Service-now.com//Outlook 11.0 MIMEDIR//EN</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>VERSION:2.0</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>METHOD:REQUEST</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>BEGIN:VEVENT</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ATTENDEE;ROLE=REQ-PARTICIPANT;RSVP=TRUE:MAILTO: ${to}</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>DTSTART:${dtstart}</td>
<td>Yes</td>
<td>You must use the import export map to map dtstart to a start time field on the custom table.</td>
</tr>
<tr>
<td>DTEND:${dtend}</td>
<td>Yes</td>
<td>You must use the import export map to map dtend to an end time field on the custom table.</td>
</tr>
<tr>
<td>UID:${sys_id}</td>
<td>Yes</td>
<td>You must provide the name of a field that uniquely identifies the record, such as the sys_id or the record number field.</td>
</tr>
<tr>
<td>DTSTAMP:${dtstamp}</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>iCalendar template line</td>
<td>Required?</td>
<td>Notes</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td>No</td>
<td>To include a multiple line description, you must use the import export map to map description to a text field on the custom table. The field mapping ensures that the system encodes line breaks correctly for the iCalendar file format.</td>
</tr>
<tr>
<td>SUMMARY:</td>
<td>No</td>
<td>To include a summary from a text field on the custom table, provide the name of the field, such as u_meeting_summary. The summary field value must not contain line breaks. You do not need to create a field mapping.</td>
</tr>
<tr>
<td>END:VEVENT</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>END:VCALENDAR</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

13. Create an email notification to trigger the iCalendar invitation and ensure the following fields are set accordingly:

- **Table**: Select the custom table.
- **Type**: Select **Meeting Invitation**.
- **Content type**: Select **Plain text** only.
- **Email template**: Select the template you created.

**Inbound email actions**

Inbound email actions enable you to define the actions an instance takes when receiving email.

Inbound email actions are similar to business rules: both use conditions and scripts that take action on a target table. An inbound email action checks the email for a watermark that associates it with a task and checks for other conditions. If the conditions are met, the inbound email action takes the action that you configure. It can take two types of actions:

- **Record action**: setting a value for a field in the target table.
- **Email reply**: sending an email back to the source that triggered the action.

By default, if an email has no identifiable watermark, an inbound email action attempts to create an incident from the message. If the email has a watermark of an existing incident, an inbound email action updates the existing incident according to the action's script.

**Attachments**

If an inbound email contains one or more email attachments, the inbound email action adds the attachments to the first record the action produces.
Character encoding

- If the email encoding is ASCII-7 or UTF-8, inbound email actions preserve the character encoding in any associated task records they produce.
- If the email encoding is ISO-8859-1, the inbound email action attempts to convert the email to Windows 1252.
- Inbound email actions convert any other encodings (for example, Mac OS Roman) to plain text, which may or may not be readable.

See the Email logs on page 1059 for examples of what you might see if a notification or inbound email action is not processed.

Domain separation

The system ignores the domain that the inbound email action record is in when it creates a record based on the inbound email action. Keep inbound actions in the global domain. For example, if your inbound email action creates an incident, the system creates the incident in the same domain as the user in the Caller field. If that user is not in the User [sys_user] table, the incident is in the global domain.

Inbound action type criteria
An instance uses a set of criteria to decide what action to take based on the type of message that is sent to the instance.
Figure 778: How an instance matches email to an inbound action type
<table>
<thead>
<tr>
<th>Inbound email action type</th>
<th>Required matching criteria</th>
<th>Name of default action (Incident table)</th>
<th>Result of default action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward</td>
<td>The email contains the following conditions:  1. A subject starting with a recognized forward prefix (even if a watermark or an In-Reply-To header is present).  2. From &lt;user email&gt; appears anywhere in the email body.</td>
<td>Create Incident (Forwarded)</td>
<td>Create new record</td>
</tr>
<tr>
<td>Reply</td>
<td>The email contains one of the following conditions and the table specified in the email matches the table of the inbound action:  1. A valid watermark that matches an existing record.  2. An In-Reply-To email header (when no watermark is present) that matches an existing record.  3. A subject line starting with a recognized reply prefix (when neither a watermark nor an In-Reply-To header is present) and a valid record number that matches an existing record.</td>
<td>Update Incident (BP)</td>
<td>Update existing record</td>
</tr>
</tbody>
</table>
Inbound email action type | Required matching criteria | Name of default action (Incident table) | Result of default action
---|---|---|---
New | The email does not meet the conditions for either a reply or forward type inbound email action | Create Incident | Create new record

If more than one inbound action is available for a particular type, the instance uses the Table field to match the email to a particular table. If there is also more than one action for the inbound action’s table, the instance uses the **Order** field to determine the order in which the actions run.

**Recognized reply prefixes**

Emails with certain prefixes trigger inbound email actions of the type **reply**.

When no watermark or is present or the **In-Reply-To** email header is present, the instance recognizes email containing a prefix from the glide.email.reply_subject_prefix property as reply email. When this property is set, your email system uses non-standard reply prefixes.

**Table 715: Specifying recognized reply prefixes**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.reply_subject_prefix</td>
<td>Specifies the list of prefixes (comma-separated) in the subject line that identify an email reply.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: re:,aw:,r:</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>

**Note:** Prefixes are case insensitive.

**Recognized forward prefixes**

Emails with certain prefixes trigger inbound email actions of the type **forward**.

The instance recognizes any email whose subject line contains a prefix from the glide.email.reply_subject_prefix property as forwarded email. Emails with these prefixes trigger inbound email actions of the type forward. When this property is set, your email system uses non-standard forward prefixes or you want forwards to behave like replies.
Table 716: Specifying recognized forward prefixes

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.forward_subject_prefix</td>
<td>Specifies the list of prefixes (comma-separated) in the subject line that identify a forwarded email.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: fw:, fwd:</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Properties [sys_properties] table</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Prefixes are case insensitive.</td>
</tr>
</tbody>
</table>

Email forwards as replies
Properties are available to force inbound actions to process forwarded mail as replied mail.

These properties control the subject prefix that the inbound actions use.

Table 717: Properties that process forwards as replies

<table>
<thead>
<tr>
<th>Property</th>
<th>Value needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email.reply_subject_prefix</td>
<td>re:, Re:, RE:, aw:, r:, fw:, fwd:, Fwd:, FWD:</td>
</tr>
<tr>
<td>glide.email.forward_subject_prefix</td>
<td>[any text that is not a forward prefix]</td>
</tr>
</tbody>
</table>

This causes all forwarded and replied to mail to be processed by the Update Incident inbound action.

**Note:** The glide.email.forward_subject_prefix property must contain some text in order for the forwarded email to be processed as a Reply. It can be any text except a forward prefix (that is, fw:, fwd:, Fwd:, FWD:).

The matching of a sender email address to a user
The instance matches a senders email address to an active user in the User [sys_user] table using inbound actions.

**Note:** The Email Automatic User Creation plugin must be active.

When processing an email, the instance sets the current user to the user whose email address matches email.from. Inbound actions can then reference that current user. For example, the base system inbound action Create Incident sets the incident's caller_id to the value returned by gs.getUserID().

If multiple users have the same email address, the instance first searches for an active user with the email address. The instance does not match inactive users.

**Note:** It is strongly recommended to have unique email addresses for each user record. Otherwise, the instance can not reliably match the email to the correct user and unpredictable matches may occur.

If providing a unique email address to each user is not possible, only having one active user with the shared email address is recommended. This configuration guarantees that the instance always matches incoming email from this address to the active user.

The matching of incoming email to an inbound action type
The instance uses the following logic to match an email to a specific inbound action type.

**Incoming email matching**

By default, the instance searches email for a watermark to match the incoming email to an existing record in the system.
The instance searches for watermarks in the email subject line first, then in the email body. If no watermark is present, the instance searches for an In-Reply-To email header that matches an existing record. If that is not found, the instances searches the subject line for a recognized reply prefix.

Note: Some versions of Microsoft Exchange Server require a hotfix to support the "In-Reply-To" header when replying to email that originates from outside the organization.
Examples of matching watermarks in the Subject line or Body
The following examples illustrate how the instance matches watermarks in an email's subject line or body.

### Table 718: Examples of matching watermarks in the Subject line or body

<table>
<thead>
<tr>
<th>Subject Line or Body Contents</th>
<th>Matching Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref:MSG0000008</td>
<td>The instance recognizes this as a watermark and searches the Email Watermarks [sys_watermarks] table for a record with the number MSG0000008. If this watermark exists, the instance matches the email to the associated record. If this watermark does not exist, the instance uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
<tr>
<td>Ref:MSGWTR0000008</td>
<td>The instance recognizes this as a watermark and searches the Email Watermarks [sys_watermarks] table for a record with the number MSGWTR0000008. If this watermark exists, the instance matches the email to the associated record. If this watermark does not exist, the instance uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
<tr>
<td>Ref:WTR0000008</td>
<td>The instance recognizes this as a watermark and searches the Email Watermarks [sys_watermarks] table for a record with the number WTR0000008. If this watermark exists, the instance matches the email to the associated record. If this watermark does not exist, the instance uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
<tr>
<td>MSG0000008</td>
<td>The instance does not recognize this as a watermark because it does not start with the Ref: prefix. It uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
</tbody>
</table>

Examples of matching record numbers in the Subject line
The following examples illustrate how the instance matches record numbers in an email's subject line to an existing record when no watermark is present.
Table 719: Examples of matching record numbers in the Subject line

<table>
<thead>
<tr>
<th>Subject line contents</th>
<th>Matching results</th>
</tr>
</thead>
<tbody>
<tr>
<td>RE: Example INC0005574</td>
<td>The instance recognizes this subject line as a reply and recognizes the INC prefix as belonging to the Incident table. The instance searches the Incident table for an existing record INC0005574. If this incident exists, the email is associated with this incident. If this incident record does not exist, the instance uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
<tr>
<td>RE: Example &quot;INC0005574&quot;</td>
<td>The instance recognizes this subject line as a reply but does recognize the &quot;INC prefix as belonging to the Incident table because of the quotation mark. The same error occurs for any character other than a space before the record number. The instance instead uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
<tr>
<td>RE: Example INC0005574*</td>
<td>The instance recognizes this subject line as a reply and recognizes the INC prefix as belonging to the Incident table. The instance searches the Incident table for an existing record INC0005574&quot;, which it will not find because of the quotation mark. The same error occurs for any character other than a space at the end of the record number. The instance instead uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
<tr>
<td>RE: CHG0008593 and INC000576</td>
<td>The instance recognizes this subject line as a reply and recognizes one, but not both, of the number prefixes. There is no way to predict which prefix the instance will match first. Whichever prefix it matches, it searches the corresponding table for a matching record. If the record exists, the email is associated with the table. If the record does not exist, the instance uses the inbound action for new emails to create an incident and associates the new incident with the email.</td>
</tr>
</tbody>
</table>

**Note:** The instance does not support processing email with multiple numbers in the subject line because there is no way to predict which record the instance will match first. For this reason, the instance does not recommend creating notifications that include more than one $number variable.
<table>
<thead>
<tr>
<th>Subject line contents</th>
<th>Matching results</th>
</tr>
</thead>
<tbody>
<tr>
<td>FW: Example INC0005574</td>
<td>The instance recognizes this subject line as a forward because of the <code>FW:</code> prefix and uses the inbound action for forwarded emails to create an incident, and associates the new incident with the email.</td>
</tr>
<tr>
<td>Example INC0005574</td>
<td>The instance recognizes this subject as a new email because it does not contain a matching reply or forward prefix. It uses the inbound action for new emails to create an incident, and associates the new incident with the email.</td>
</tr>
</tbody>
</table>

*Create an inbound email action*

You can create inbound email actions to define the actions that the system takes when an email is received.

*Role required: admin*

1. Navigate to **System Policy > Email > Inbound Actions**.
2. Click **New**.
3. Fill in the fields as described in the table.
Inbound Email Actions - Create Incident

Inbound email actions specify how ServiceNow creates or updates task records in a table when the instance receives an email. The inbound email action looks for a watermark in the email to associate it with a specific task. If the conditions specified in the inbound action are met, the script is run. More info

| Name       | Create Incident       | Application | Global
|------------|-----------------------|-------------|--------
| Target table | Incident [Incident]  | Active      |        
| Action type | Record Action         | Stop processing |     

When to run

<table>
<thead>
<tr>
<th>Only emails of the selected Type will trigger this inbound action.</th>
<th>Only emails from senders with the Required roles will trigger this inbound action.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type New</td>
<td>Required roles</td>
</tr>
</tbody>
</table>

Order determines when to run relative to other inbound actions. The inbound action with the lowest order runs first.

<table>
<thead>
<tr>
<th>Order</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Only emails from this sender will trigger this inbound action.

All of the following conditions must be true, to trigger this inbound action.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Add Filter Condition</th>
<th>Add &quot;OR&quot; Clause</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- choose field --</td>
<td>-- oper --</td>
<td>-- value --</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th></th>
</tr>
</thead>
</table>
Note: You might need to configure the form to see all fields.

Table 720: Inbound Email Actions form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When to run</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Enter a descriptive name for this email action.</td>
</tr>
<tr>
<td>Target table</td>
<td>Select the table where the action will add or update records.</td>
</tr>
<tr>
<td>Action type</td>
<td>Select the type of action the instance takes. Select <strong>Record Action</strong> to modify a record in the instance or select <strong>Reply Email</strong> to have the instance send an email back to the source of the inbound email.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to prevent the system from running additional inbound email actions after this one runs. Clear the check box to disable the action.</td>
</tr>
<tr>
<td>Stop processing</td>
<td>Select this option to stop processing after this inbound email action runs.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the message type required to run the action. The action runs only if the inbound email is of the selected type. Available types are:</td>
</tr>
<tr>
<td></td>
<td>• New: An email that is not recognized as a reply or forward.</td>
</tr>
<tr>
<td></td>
<td>• Reply: An email with a watermark with an In-Reply-To email header, or whose subject line begins with a recognized reply prefix.</td>
</tr>
<tr>
<td></td>
<td>• Forward: An email whose subject line begins with a recognized forward prefix, even if the email also contains a watermark or In-Reply-To header.</td>
</tr>
<tr>
<td>Required roles</td>
<td>Specify required roles the sender must have to trigger the inbound action.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Order</td>
<td>Enter a number that specifies when this inbound action runs relative to other inbound actions that use the same target table. The instance processes the action with the lowest order number first.</td>
</tr>
<tr>
<td>From</td>
<td>Select the user required to run the action. If a user is selected, the action runs only when the email sender matches the user name. Leave this field blank to perform the action for all users.</td>
</tr>
</tbody>
</table>
| Condition     | Specify the condition that must evaluate to true to trigger the inbound action. Build a condition with the choice lists or enter a statement that determines when the inbound email action runs. For example:  

```java
email.subject. startsWith ( "chg:" )
```

Actions

<table>
<thead>
<tr>
<th>Field actions</th>
<th>Specify how information in the email is inserted into the record. For example, select [Created by] [From email] [Sender], so that when the request is inserted, you can see who it is for. This field appears if the action type is Record Action.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply email</td>
<td>Compose the email message to send to the source that triggered the inbound email action.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Script</td>
<td>Enter the script the action runs. Typically, this script uses the validators script include and email variables. A template is provided:</td>
</tr>
<tr>
<td></td>
<td><code>(function runAction(/ *GlideRecord*/ current, / *GlideRecord*/ event, / *EmailWrapper*/ email, / *ScopedEmailLogger*/ logger) { // Implement email action here }) (current, event, email, logger);</code></td>
</tr>
<tr>
<td></td>
<td>These are the objects available:</td>
</tr>
<tr>
<td></td>
<td>• <strong>current</strong>: access the record referred to by the inbound email. For example, <code>current.assigned_to</code> accesses the person assigned to the task.</td>
</tr>
<tr>
<td></td>
<td>• <strong>event</strong>: access one of the parameters of the originating event. For example, <code>event.parm1</code> accesses the first parameter of the event or <code>event.parm2</code> for the second parameter. See <strong>Events</strong> on page 2996 for more information.</td>
</tr>
<tr>
<td></td>
<td>• <strong>email</strong>: access the inbound current email record. For example, <code>email.subject</code> accesses the content in the subject line of the email. See <strong>Accessing email objects with variables</strong> on page 2837 for more information.</td>
</tr>
<tr>
<td></td>
<td>• <strong>logger</strong>: add a message to the log file with the source set to <code>email.&lt;Sys ID of incoming email&gt;</code>. For example: <code>logger.log (&quot;Some information&quot;)</code></td>
</tr>
</tbody>
</table>
| Description | Enter a detailed explanation of what this inbound email action does.

Other fields
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>Enter a number to define the order in which this email action should be processed. Actions with lower numbers are processed first. For more information. This field is not installed by the Ordered Email Processing plugin.</td>
</tr>
</tbody>
</table>

**Accessing email objects with variables**

An inbound email action script has access to various pieces of an inbound email through script variables.

**Table 721: Accessing email objects with variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>email.to</td>
<td>Contains a comma-separated list of email addresses in the To: and Cc: boxes.</td>
</tr>
<tr>
<td>email.direct</td>
<td>email.direct</td>
</tr>
<tr>
<td>email copied</td>
<td>Contains a comma-separated list of email addresses in the Cc: box.</td>
</tr>
<tr>
<td>email.body_text</td>
<td>Contains the body of the email as a plain text string.</td>
</tr>
<tr>
<td>email.body_html</td>
<td>Contains the body of the email as an HTML string.</td>
</tr>
<tr>
<td>email.from</td>
<td>Contains an email address that depends on the following conditions:</td>
</tr>
<tr>
<td></td>
<td>• If the address listed in the email Headers field matches an existing user’s Email address, this variable contains the user’s email address.</td>
</tr>
<tr>
<td></td>
<td>• If the address listed in the email Headers field does not match an existing user’s Email address, this variable contains the address listed in the email Headers field.</td>
</tr>
<tr>
<td>email.from_sys_id</td>
<td>Contains the Sys ID of the user who sent the email to the instance.</td>
</tr>
<tr>
<td>email.origemail</td>
<td>Contains the email sender’s address as listed in the email Headers field.</td>
</tr>
<tr>
<td>email.subject</td>
<td>Contains the email’s subject as a plain text string.</td>
</tr>
<tr>
<td>email.recipients</td>
<td>Contains a comma-separated list of recipient addresses.</td>
</tr>
<tr>
<td>email.recipients_array</td>
<td>Contains the recipient addresses as an array.</td>
</tr>
<tr>
<td>Variable</td>
<td>Contents</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>email.content_type</td>
<td>Contains the MIME content type of the email (for example, text/plain; charset=&quot;us-ascii&quot; or text/html; charset=&quot;us-ascii&quot;).</td>
</tr>
<tr>
<td>email.headers</td>
<td>Contains details about the sender, route, and receiver as a plain text string in the format of the sending email client.</td>
</tr>
</tbody>
</table>

**Note:** The instance follows RFC 2822 (Internet Message Format), which requires multiple email addresses in a group to be separated by commas, not semicolons. The instance can set the values of the email.to, email.direct, and email.copied variables only if email addressed to groups follow the expected RFC format.

**Example assignment of tasks via email**

Name:value pairs in an email template can set field values in a record.

For example, the `change.itil.approve.role` email template lists several fields in the outbound notification.

![Email Template](image)

**Figure 781: The Email Template form**

Notice the line in the template that shows `Priority:${sysapproval.priority}`. When replying to this email, the approver can change the value of the **Priority** field directly from the email. For example, the approver could set the priority to 4—Low:

```
Priority: 4
```

**Locked out users triggering inbound email actions**

A property is available to have locked out users trigger inbound actions.

The property `glide.pop3.process_locked_out` allows users who need to reset their password to send an email message to an instance asking for assistance.

**Warning:** Enabling this property also enables users from untrusted domains to trigger inbound actions.
Table 722: Property that allows locked out users to trigger inbound email actions

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.pop3.process_locked_out</td>
<td>Enables (true) or disables (false) the ability for locked out users to trigger inbound actions.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: false</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>

**Email user matching**

When the instance receives an email message, the system searches for an existing user record with the same email address as the sender.

Table 723: Matching email to existing users

<table>
<thead>
<tr>
<th>Value of email.from Variable</th>
<th>Matching User ID</th>
<th>Email Address</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td><a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td><a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td>Michael Tossi</td>
</tr>
<tr>
<td>&quot;Michael Tossi&quot;<a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td><a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td><a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td>Michael Tossi</td>
</tr>
<tr>
<td>&quot;Tossi, Michael&quot;<a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td><a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td><a href="mailto:michael.tossi@company.com">michael.tossi@company.com</a></td>
<td>Michael Tossi</td>
</tr>
<tr>
<td>&quot;Tossi&quot;<a href="mailto:mtossi@company.com">mtossi@company.com</a></td>
<td><a href="mailto:mtossi@company.com">mtossi@company.com</a></td>
<td><a href="mailto:mtossi@company.com">mtossi@company.com</a></td>
<td>Tossi</td>
</tr>
</tbody>
</table>

**Note:** This functionality requires that you activate the Email Automatic User Creation plugin.

Inbound Email Action scripts no longer support the gs.createUser() method. Use either the automatically-generated email variables or the gs.GetUserID() method instead.

User impersonations and inbound actions

When the instance receives an email, it can take a variety of actions by impersonating the sender.

If the sender of an incoming email matches an existing user, the instance impersonates the matching user to complete any inbound email actions. If the sender does not match an existing user, the instance impersonates the Guest user to complete any inbound email actions. If the impersonated user is locked out, the inbound email action fails.

**Note:** If inbound email comes from an untrusted domain, the instance impersonates the Guest user unless you explicitly prevent users from untrusted domains from triggering inbound actions.

**User creation from incoming email**

An instance can automatically create users from incoming email.

When an instance receives a message and there is no matching email address from the sender, the instance can create a user with the User ID [sys_user.user_name] set to the sender’s full email address.
Table 724: Creating users from incoming email

<table>
<thead>
<tr>
<th>Value of email.from Variable</th>
<th>User ID Created</th>
<th>Email Address</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td><a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td><a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td>New User</td>
</tr>
<tr>
<td>&quot;New User&quot; <a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td><a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td><a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td>New User</td>
</tr>
<tr>
<td>&quot;User, New&quot; <a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td><a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td><a href="mailto:new.user@company.com">new.user@company.com</a></td>
<td>New User</td>
</tr>
<tr>
<td>&quot;User&quot; <a href="mailto:nuser@company.com">nuser@company.com</a></td>
<td><a href="mailto:nuser@company.com">nuser@company.com</a></td>
<td><a href="mailto:nuser@company.com">nuser@company.com</a></td>
<td>User</td>
</tr>
</tbody>
</table>

Enabling automatic user creation

To enable the automatic creation of user records from email, you must set two email properties.

1. Navigate to **System Properties > Email**.
2. Select the check box for **Automatically create users for incoming email from trusted domains** (glide.pop3readerjob.create_caller).
3. Enter the list of trusted domains in **Trusted domains for creating users from incoming emails** (glide.user.trusted_domain).
4. Click **Save**.

**Note:** When the property glide.pop3readerjob.create_caller is set to false, the instance runs inbound actions from users who do not match an existing user by impersonating the Guest user. When the property glide.pop3readerjob.create_caller is set to false, the instance runs inbound actions from users who do not match an existing user by impersonating Guest user. You may want to prevent untrusted users from triggering inbound actions by locking out the Guest user.

User creation method

The method the instance uses to create users can be upgraded to use the full email address by activating the Email Automatic User Creation plugin.

The plugin makes the following changes:

- Sets the property glide.email.create_userid_from_email to true.
• Increases the width of the User ID \([\text{sys_user.user_name}]\) column to accommodate email addresses.

After activating the plugin, enable automatic user creation from email.

**Warning:** Review your existing user records to reconcile any that contain identical email addresses. If you activate the plugin prior to reconciling email addresses, your instance cannot distinguish between users with identical email addresses and will randomly select one of the users with the matching email address.

prevent untrusted users from triggering inbound actions
You can prevent users from untrusted domains from triggering inbound actions.

Role required: admin

For example, you can prevent email from users outside your company domain from creating incidents.

**Note:** Users in your instance must still have write and update access to the records that they create or update through inbound email actions.

1. Enable automatic user creation and add a list of trusted domains. For example, add your company domain `example.com`.
2. Navigate to **User Administration > Users**.
3. Select the user `guest`.
4. Select the **Locked out** field to disable the guest account. Locking out a user record prevents the user from processing inbound actions.

When a user from a trusted domain sends an email to the instance, the instance either matches the email to an existing user or creates a new user. Since the incoming email matches a user record (either an existing or new one), the email can trigger an inbound action.

When a user from an untrusted domain sends an email to the instance, the instance attempts to impersonate the guest user. Since the guest user is locked out, the impersonation fails and the incoming email cannot trigger an inbound action.

**Warning:** Allowing locked out users to trigger inbound actions also allows untrusted users to trigger inbound actions.

inbound email recipient processing
The recipients variables allow processing of inbound email based on the email recipients.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>email.recipients</code></td>
<td>Returns a comma-separated list of recipient addresses as a plain text string.</td>
</tr>
<tr>
<td><code>email.recipients_array</code></td>
<td>Returns the recipient addresses as an array.</td>
</tr>
</tbody>
</table>

For example, create a script to process email based on the array values:

```javascript
var rarray  = email.recipients_array ; for ( var i  = 0 ; i  < rarray.length ; i ++ ) { var recipient  = rarray [i ] ; // do something with it }
```

© 2017 ServiceNow. All rights reserved.
The sys_email variable
You can use the global variable sys_email with inbound email actions.

This lets you access the received sys_email record that triggered the inbound email action. It can be used to reference fields on the email record, such as uid, sys_id, content_type, and so on.

Email redirection to the instance POP3 account
You can have other mailboxes forward email to the instance’s POP3 account.

By default, the POP Reader schedules job checks for new email every two minutes. It connects to the mail server and account specified in email properties. The POP Reader downloads any email waiting on the mail server and creates email.read events. After the instance processes the events, the inbound email actions run.

The POP Reader shows the number of emails processed during the reader’s last run. The message shows the number of emails the reader processed or 0 processed if no emails were available. The reader resets the status each time it runs.

While it is not possible to specify more than one POP3 account for the instance, you can have other mailboxes forward to the designated POP3 account. This script can be added to the Create Incident inbound email action to differentiate the content based on the original recipient and then set an assignment_group value.

```javascript
if(email.direct.indexOf('facilities@anycorp.com')>-1)
current.assignment_group.setDisplayValue('Facilities Management');
```

Field values from the email body
Values in an inbound email can set field values in a task record.

Any name:value pair in an inbound email body gets parsed into a variable/value pair in the inbound email script. The name:value pair must be on its own line. Note that most email clients limit the number of characters allowed per line and may truncate excessively long name:value pairs.

To populate a reference field, use setDisplayValue() instead. See Redirecting Emails for an example of using setDisplayValue() in an inbound email action.

Note: The action always generates a lowercase variable name. Also, this functionality does not work on reference fields.

For example, if an email body contains this line:

```
Foo:bar
```

The inbound email script creates the variable email.body.foo with the value of bar. You can use these variables to create conditions such as:

```javascript
if(email.body.foo!=undefined){
  current.[field]=email.body.foo;
}
```

In this example, the script sets the value of [field] to the value bar.

Integrating inbound events
This tutorial illustrates how to create a notification from an inbound JSON request.

Role required: admin

When complete, you will be able to:

- Send a JSON request to the imp_notification web service import set with the JSON processor
- Create a new import set in the imp_notification table in the instance using data from the JSON request
The following example steps assume you have your own demonstration instance.

1. Activate the JSON Web Service plugin.
2. Install the RESTClient Firefox plugin.
3. Open the RESTClient.
4. Create the following JSON request.
   - **Method**: POST
   - **URL**: http://<instance name>.service-now.com/imp_notification.do?JSON
   - **Headers**: Authorization: Basic
   - **Body**: 
     ```
     {"sysparm_action":"insert","message":"this is an event","uuid":"abc"}
     ```

5. Click **Send**.
7. Verify that the instance sends back a response with a **sys_id**.
8. Login to your development instance.


10. Verify that the import set table has an event matching your JSON request.

**Troubleshooting inbound email**

The following solution to this common inbound email error is useful for troubleshooting email actions.

**Table 726: Issues and symptoms**

<table>
<thead>
<tr>
<th>Error or symptom</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email from Outlook produces an empty Incident record</td>
<td>Configure the local Outlook client or Exchange server to not send Rich Text formatted (RTF) data to the instance. To prevent Windows email containing a winmail.dat file, see Microsoft <a href="https://support.microsoft.com/en-us/help/138053">KB 138053</a> for information on configuring Exchange.</td>
</tr>
</tbody>
</table>

For more information, open the Knowledge Base and access the following two articles:

- For troubleshooting inbound email, see [KB0524472](https://knowledge.citrix.com/knowledgebase/view.php3?id=kb0524472).
- For troubleshooting outbound email, see [KB0524480](https://knowledge.citrix.com/knowledgebase/view.php3?id=kb0524480).

**Inbound email action examples**

Several examples of inbound email actions are available to help you build your own inbound email actions. Inbound email action example: handling email replies
This example shows you how to set up inbound email actions to handle replies that users send back to the instance.

Role required: admin

The inbound email action parses the email and responds using a script. By default, an email received by the instance creates a new incident, and the body of the email is added to the Additional Comments text box. More refined Inbound Email Actions can create incident tickets with more data, thus saving the incident management team valuable time.

Normally, when a user responds to an email sent by the instance, the inbound email action matches the watermark to an existing incident, and updates the incident rather than creating a new record. However, if the watermark is missing, this inbound email action attempts to match a reply to the original incident.

1. Navigate to System Policy > Inbound Actions and click New.
2. Populate the form as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Update Incident</td>
</tr>
<tr>
<td>Type</td>
<td>Reply</td>
</tr>
<tr>
<td>Target table</td>
<td>Incident [incident]</td>
</tr>
</tbody>
</table>

3. In Script, enter this code.

```javascript
gs.include('validators');

//Note: current.caller_id and current.opened_by are already set to the first UserID that matches the From: email address
if (current.getTableName() == "incident") {
    current.comments = "reply from: " + email.origemail + "\n\n" + email.body_text;
    if (email.body.assign !== undefined)
        current.assigned_to = email.body.assign;
    if (email.body.priority !== undefined && isNumeric(email.body.priority))
        current.priority = email.body.priority;
    if (email.body.category !== undefined)
        current.category = email.body.category;
    if (email.body.short_description !== undefined)
        current.short_description = email.body.short_description;

    current.update();
}
```

Inbound email action example: logging a problem

This example shows you how to set up inbound email actions to create a problem record.

Role required: admin

Inbound email actions allow users to log or update incidents on an instance via email. The inbound email action parses the email and responds using a script.

1. Navigate to System Policy > Inbound Actions and click New.
2. Populate the form as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Log Problem</td>
</tr>
<tr>
<td>Type</td>
<td>New</td>
</tr>
<tr>
<td>Active</td>
<td>True</td>
</tr>
<tr>
<td>Target Table</td>
<td>Problem [problem]</td>
</tr>
<tr>
<td>Condition</td>
<td>email.subject.indexOf(&quot;Problem: &quot;) == 0</td>
</tr>
<tr>
<td>Script</td>
<td>current.description =</td>
</tr>
<tr>
<td></td>
<td>email.body_text;</td>
</tr>
<tr>
<td></td>
<td>current.short_description =</td>
</tr>
<tr>
<td></td>
<td>email.subject.toString().substring(9);</td>
</tr>
<tr>
<td></td>
<td>current.assignment_group.setDisplayValue(&quot;Development&quot;);</td>
</tr>
<tr>
<td></td>
<td>if (email.body.assign != undefined)</td>
</tr>
<tr>
<td></td>
<td>current.assigned_to =</td>
</tr>
<tr>
<td></td>
<td>email.body.assign;</td>
</tr>
<tr>
<td></td>
<td>current.insert();</td>
</tr>
</tbody>
</table>

Inbound email action example: requesting a change

This example shows you how to set up inbound email actions to create a change request record.

Role required: admin

Inbound Email Actions allow users to log or update incidents on an instance via email. The inbound email action parses the email and responds using a script.

1. Navigate to System Policy > Inbound Actions and click New.
2. Populate the form as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Request Change</td>
</tr>
<tr>
<td>Type</td>
<td>New</td>
</tr>
<tr>
<td>Active</td>
<td>True</td>
</tr>
<tr>
<td>Target Table</td>
<td>Change Request [change_request]</td>
</tr>
<tr>
<td>Condition</td>
<td>email.subject.indexOf(&quot;Change Request: &quot;) == 0</td>
</tr>
<tr>
<td>Script</td>
<td>current.comments =</td>
</tr>
<tr>
<td></td>
<td>email.body_text;</td>
</tr>
<tr>
<td></td>
<td>current.short_description =</td>
</tr>
<tr>
<td></td>
<td>email.subject;</td>
</tr>
</tbody>
</table>
Inbound email action example: updating an incident

Inbound Email Actions allow users to log or update incidents on an instance via email. The Inbound Email Action parses the email and responds using a script.

Role required: admin

By default, an email received by the instance creates a new incident, sets the Contact type field to Email, and adds the body of the email to the Additional Comments field. More refined Inbound Email Actions can create incident tickets with more data, thus saving the incident management team valuable time.

The following Inbound Email Action applies to email replies. Normally, when a user responds to an email sent by the instance, the inbound email action will match the watermark to an existing incident, and update the incident rather than creating a new record. However, if the watermark is missing, this Inbound Email Action will attempt to match a reply to the original incident.

To define an inbound email action for replies:

1. Navigate to System Policy > Email > Inbound Actions and click New.
2. Populate the form (see table).

<table>
<thead>
<tr>
<th>Field</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>current.notify = 2;</td>
<td></td>
</tr>
<tr>
<td>if (email.body_text.assign !== undefined)</td>
<td></td>
</tr>
<tr>
<td>current.assigned_to = email.body_text.assign;</td>
<td></td>
</tr>
<tr>
<td>if (email.body_text.priority !== undefined)</td>
<td></td>
</tr>
<tr>
<td>current.priority = email.body_text.priority;</td>
<td></td>
</tr>
<tr>
<td>if (email.body_text.category !== undefined)</td>
<td></td>
</tr>
<tr>
<td>current.category = email.body_text.category;</td>
<td></td>
</tr>
<tr>
<td>current.insert();</td>
<td></td>
</tr>
</tbody>
</table>

Inbound Email Actions allow users to log or update incidents on an instance via email. The Inbound Email Action parses the email and responds using a script.

Role required: admin

By default, an email received by the instance creates a new incident, sets the Contact type field to Email, and adds the body of the email to the Additional Comments field. More refined Inbound Email Actions can create incident tickets with more data, thus saving the incident management team valuable time.

The following Inbound Email Action applies to email replies. Normally, when a user responds to an email sent by the instance, the inbound email action will match the watermark to an existing incident, and update the incident rather than creating a new record. However, if the watermark is missing, this Inbound Email Action will attempt to match a reply to the original incident.

To define an inbound email action for replies:

1. Navigate to System Policy > Email > Inbound Actions and click New.
2. Populate the form (see table).

<table>
<thead>
<tr>
<th>Name</th>
<th>Update Incident</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action type</td>
<td>Reply Email</td>
</tr>
<tr>
<td>Target table</td>
<td>Incident [incident]</td>
</tr>
<tr>
<td>Actions &gt; Script</td>
<td>Insert the following:</td>
</tr>
</tbody>
</table>

```javascript
gs.include('validators');
if (current.getTableName() == "incident") {
    current.comments = "reply from: " + email.origemail + "\n\n" + email.body_text;
    if (email.body.assign !== undefined)
        current.assigned_to = email.body.assign;
```
if (email.body.priority !== undefined && isNumeric(email.body.priority))
  current.priority = email.body.priority;

if (email.body.category !== undefined)
  current.category = email.body.category;

if (email.body.short_description !== undefined)
  current.short_description = email.body.short_description;

  current.update();
}

Values automatically set from incoming email

The default inbound action for the Incident table automatically sets the following field values when it receives an incoming email.

<table>
<thead>
<tr>
<th>Field value set</th>
<th>Value used from incoming email</th>
</tr>
</thead>
<tbody>
<tr>
<td>current.caller_id</td>
<td>User ID of the first user whose email address matches the <code>email.from</code> variable.</td>
</tr>
<tr>
<td>current.opened_by</td>
<td>User ID of the first user whose email address matches the <code>email.from</code> variable.</td>
</tr>
</tbody>
</table>

If multiple users have the same email address, the instance first searches for an active user with the email address. The instance recommends having unique email addresses for each user record. If providing a unique email address to each user is not possible, the instance recommends only having one active user with the shared email address. This configuration guarantees that the instance always matches incoming email from this address to the active user.

**Ordered email processing**

The Ordered Email Processing plugin enables you to configure a processing order for inbound email actions.

In addition to forcing email actions to run in a prescribed order, the administrator can add a command to the script for an action that halts processing after the script runs. The plugin simply adds the `Order` column to the Rules [sysrule] table, which the instance uses to determine when to process emails.

Configure the processing order

Configure the processing order for inbound email actions to force them to run in a prescribed order.

Role required: admin

1. Navigate to **System Policy > Email > Inbound Actions**.
2. Open an existing inbound action or create one.
3. Complete the form and assign an order number to the `Order` field to establish when this inbound rule should run in relation to other rules. If you upgraded and activated the plugin, the `Order` field might be named **Execution Order**.
**Note:** Ensure each inbound action has a unique **Order** value to ensure the system stops processing as expected. If multiple inbound actions have the same **Order** value, the system might evaluate all of the inbound actions, even if one of them contains the `event.state="stop_processing";` script or has the **Stop processing** option selected.

4. To stop rule processing when an inbound email action runs successfully: add the following line to the bottom of the script:

- Select the **Stop processing** check box.
- Add the following line to the bottom of the **Actions** script:

```
event.state="stop_processing";
```
Manage multiple filters in an inbound email action

Use process ordering and the `stop_processing` command to manage multiple filters in inbound email actions.

Role required: admin
In this example, you can create new problem records when `prb:` appears in the subject line and new change requests when `chg:` appears in the subject line. All other emails are used to create an incident. The actions are set up as follows:

1. Create an action with a condition of `Subject > starts with > chg:` and the `event.state="stop_processing";` command appended to the script.
2. Assign this action an Order value of 100.
3. Create an action with a condition of `Subject > starts with > prb:` and the `event.state="stop_processing";` command appended to the script.
4. Assign this action an Order value of 200.
5. Create an action for incident with no conditions and an Order value of 300.

It is not necessary to add the `stop_processing` command to the script for the incident action unless you want processing to stop at this rule to avoid continuing to another action.

If either a change request or a problem is created, the `stop_processing` command stops processing, and no incident record is created. If neither a change request nor a problem is created, the inbound email action for incident creates a record.

Activate ordered email processing

The Ordered Email Processing plugin is enabled by default for all new instances and can be installed by a user with the admin role for upgraded instances.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click Activate.

Email filters

Email filters enable administrators to specify when to move email to particular mailboxes or to ignore it using a condition builder or a condition script. Email filters are inactive by default.

Default email filters

After activation, these filters are available from the System Mailboxes > Filters module:

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore VCAL</td>
<td>Ignores all email containing vCalendar requests. This filter prevents inbound email actions from creating unnecessary incident records when the instance receives a response to sent email. vCalendar requests in email responses are identified by the EmailUtils script include.</td>
</tr>
</tbody>
</table>
### Filter Description

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore header</td>
<td>Ignores email that contains specific headers. This filter overrides the <code>glide.pop3.ignore_headers</code> property.</td>
</tr>
<tr>
<td>Ignore sender</td>
<td>Ignores email from specific senders. This filter overrides the <code>glide.pop3.ignore_senders</code> property.</td>
</tr>
<tr>
<td>Ignore subject</td>
<td>Ignores email with specific terms or phrases in the subject line. This filter overrides the <code>glide.pop3.ignore_subjects</code> property. This filter might not apply to emails arriving from unknown users. Unknown users can be locked out.</td>
</tr>
<tr>
<td>Move spam to junk folder</td>
<td>Moves email identified as spam to the Junk folder. This filter changes an email's x-headers to mark a message as spam. This filter does not include the software product of the same name. This filter is enhanced in the Geneva release to provide better spam filtering.</td>
</tr>
</tbody>
</table>

**Note:** Using the Move spam to junk folder filter requires configuring your own email servers as the ServiceNow email infrastructure does not support changing email headers. You can accomplish such filtering with any software capable of changing email headers.

---

**Email filter script include**

Email filters use a script include called `EmailUtils` that contains a simple utility function to determine if `vCalendar` is in the body of the response email. The results of this query are used in a condition script in the Ignore VCAL email filter.

**Spam scoring and filtering**

Every message sent through email servers is assessed for the likelihood of being spam. Based on this assessment, the instance adds headers to each message that can be used for filtering within the customer instance using the Email Filters plugin. This is only available for instances that use the ServiceNow email infrastructure. See [KB0549426](#) for an explanation of email scoring and filtering.

**Activate email filters**

Administrators can activate the Email Filters plugin (`com.glide.email_filter`).

Role required: admin

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   
   If the plugin depends on other plugins, these plugins are listed along with their activation status.

3. Optional: If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Create an email filter
You can create email filters to apply a custom action script or filter actions when email matches your filter's conditions.

Role required: admin

1. Navigate to **System Mailboxes > Administration > Filters**.
2. Click **New**.
3. Complete the fields (see the table).
4. Right-click in the header bar and select **Save** from the context menu.
   The Filter Actions related list appears.
5. Click **New** in the related list.
6. Select how the system should react when the conditions of this filter evaluate to true.
   The two choices in the base system are: Mark as Ignored and Move to Junk. You can use the Action script to perform additional email tasks.
Evaluates condition based on results of a script include

Additional tasks to perform when the filter condition evaluates to true.
### Table 730: New filter table

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Brief, descriptive name for this filter.</td>
</tr>
<tr>
<td>Order</td>
<td>Define an order for this filter to determine when it is evaluated. By default, the Ignore VCAL filter has the lowest order number and is evaluated first.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box (true) to enable this filter.</td>
</tr>
<tr>
<td>Short description</td>
<td>Concise description of the function of this filter.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Add conditions for this filter with the condition builder or with a condition script. Note that conditions are case sensitive.</td>
</tr>
<tr>
<td>Condition script</td>
<td>Create a script that defines the conditions for this filter. Use this script to access other tables, variables, or methods. For example, the condition script for the Ignore VCAL filter looks for the results of the EmailUtils script include, which detects vCalendar attachments.</td>
</tr>
<tr>
<td>Action script</td>
<td>Use the action script to define additional behavior to be performed when the filter condition evaluates to true. This might include an email action not included in the default filter actions. In the base system, the action script is used to produce the error string that is published to the email log (System Logs &gt; Email).</td>
</tr>
</tbody>
</table>

**Note:** Before creating additional email actions with a script, see *Inbound Email Actions*. Like business rules, inbound email actions use both conditions and scripts and can provide a number of useful actions on emails the instance receives.

---

### Watermarks on notification emails

By default, the system generates a watermark label at the bottom of each notification email to allow matching incoming email to existing records.

The watermark always includes "Ref:" and a customizable prefix, followed by the auto-numbered identifier of the source record, such as incident, problem, or change request. The default prefix is MSG. For example, Ref:MSG3846157.

Watermarks are always generated, but you can configure them to:

- Create a custom watermark prefix for each instance to prevent accidentally triggering events in the wrong instance.
- Have custom prefix characters after MSG
- Be hidden globally
• Be omitted from individual email messages

Inbound email actions might not work properly if watermarks are omitted from email notifications. Without a watermark, inbound email messages cannot be associated with the accumulated comments related to the original incident, and each subsequent notification that is sent is treated as a new incident.

**Note:** Do not use colons (:) in custom watermark prefixes. Colons are a reserved character and may cause the watermark to be ignored.

**Note:** Email clients that use the plain text version of the email still show the watermark.

*Create a custom watermark prefix for email notifications*

By default, email notifications use the watermark prefix **MSG**, but you can create a custom watermark prefix.

Role required: admin

Any email notifications that are forwarded from one instance to another might be indistinguishable because they use the same watermark. To avoid unintentionally triggering events in the wrong instance, create a unique watermark prefix for each instance.

1. Navigate to **System Definition > Number Maintenance**.
2. Open the **sys_watermark** record.
3. Make the **Prefix** unique for this instance.

![](image)

4. Click **Update**.

*Omit an email notification watermark*

You can omit watermarks on email notifications if you do not want the instance to match the notification to an existing record.

Role required: admin

When incoming email does not contain a watermark, the system searches the subject line and message body for an incident number. The system attempts to match any incident number that it finds to an existing
incident. If there is a matching incident number, the system updates the incident with the values in the incoming email.

1. Navigate to System Notification > Email > Notifications.
2. Select the email notification to update.
3. Click the Advanced View related link.
4. In the What it will contain section, select the Omit watermark check box.
5. [Optional] If you want response email messages to generate new incidents, remove the record number \${number} variable from the Subject and Message HTML fields.

Hide email watermarks globally
Rather than omitting watermarks, it is possible to hide watermarks on a global basis using HTML markup.

Role required: admin
1. Navigate to sys_properties.list in the Application Navigator.
2. Create a new property named glide.email.watermark.visible and set it to false.

   This ensures that all watermarks are hidden on all email messages. This cannot be done on a per-email basis.

System mailboxes
Email messages can be seen in the System Mailboxes menu, which gives you access to the system Inbox, Outbox, and Sent mailbox.

Each provides a filtered view of emails to let you see only what you need to at any given time, including list fields applicable to each. The system mailboxes menu shows your current POP and SMTP status.

The System Mailbox is hosted by ServiceNow, who have sole access to the mailboxes. By default, once the instance pulls an email message, it is deleted from the mail server and stored in the application on the Email [sys_email] table.

System Mailboxes

- **Inbound emails**: All inbound mail is placed into the Inbox until it is processed. After it is cleared, the email moves to the Received state. If the email message matches the criteria in an inbound email action, the email is changed to Processed. If not, it is changed to Ready. If the system is restarted for any reason (such as during a system upgrade), all inbound mail waits on the external mail server until the system can request delivery.

- **Outbound emails**: All outbound mail is placed into the Outbox until it is processed. Once cleared, it is moved to Sent (if sent) or Skipped (not sent, as in the case of no valid recipients). If the system is restarted for any reason (such as during a system upgrade), all outbound mail waits in the instance database until the system comes online, and the scheduler looks for mail to deliver.

The email client
The email client enables you to send email directly from any record, such as an incident, change request, problem, or user record.

The email client is enabled by default, starting with the Fuji release.

The email client can be useful in cases where you want to send an email:

- To engage more people in a piece of work
- That includes Cc and Bcc recipients
- That includes personalized comments
• To a third party who doesn't have an account in your instance
• To someone about an incident where you don't have an email notification set up to do exactly what you want

The email client always uses the **multipart/mixed** content type and supports HTML markup in the message body.

**Note:** Line breaks do not appear for multi-line fields such as `{{description}}` and `{{comments}}` in the email client template.

**Email client interface**
The instance's email client interface looks like a standard email interface, which contains a toolbar for text formatting and adding attachments.

Users see an email icon based on the UI version.

• **UI16:** The email icon (✉️) appears in the more options menu.
• **UI15:** The email icon (✉️) appears in the form header.

Users click the email icon to launch the email client as a pop-up window.
Figure 783: The email client interface

**Note:** The Subject field on the email client allows a larger character count than the default setting for the Subject field on the Email table. If the subject text from the client is being truncated, increase the Max length value for the Subject field on the Email table.

**Customize the email client**
The email client has default properties and values that you can customize to suit your needs.
<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>How to Customize</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL access</td>
<td>Defaults to allowing only users with the itil role to access the email client.</td>
<td>To control who can see the email client, see Control access to the email client on page 2862.</td>
</tr>
<tr>
<td>Email icon</td>
<td>Allows only users with write access to the current table to see the email icon.</td>
<td>To remove the email icon, see Email icon display on page 2868.</td>
</tr>
</tbody>
</table>
| Autocomplete | Defaults to only displaying a user's first and last name. A system property controls what columns the email client auto-complete displays. If your organization has several users with the same name, consider adding company or email address columns to help distinguish between users. | • To change search behavior, see Configure email client auto-complete search results on page 2860.  
• To display additional information, see Additional information in the email client auto-complete on page 2861 |
| To        | Defaults to the email address of the caller.                                  | To edit the default value, create an email client template.                                                                                                                                               |
| Cc         | Defaults to the email addresses of the user who opened the incident and all users in the watch_list. | To edit the default value, create an email client template.                                                                                                                                               |
| Subject    | Defaults to the incident number and short description.                        | To edit the default value, create an email client template.                                                                                                                                               |

Configure email client auto-complete search results
You can control the email client's auto-complete search results with an email client property.

The following properties on System Properties &gt; UI Properties control the autocomplete functionality for the email client.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of autocomplete matches returned to the Email Client. Applies separately to users and groups.</td>
<td>10</td>
</tr>
<tr>
<td>Include groups in Email Client autocomplete results</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 784: Email client autocomplete properties
### Table 732: Email client properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.email_client.autocomplete.count</td>
<td>Set the maximum number of auto-complete matches.</td>
</tr>
<tr>
<td>glide.ui.email_client.autocomplete.group</td>
<td>Specify whether groups are included in auto-complete results.</td>
</tr>
</tbody>
</table>

Additional information in the email client auto-complete

A system property allows the email client to display additional columns from the User table in the auto-complete list.

Administrators can add this system property to help distinguish between individuals who have the same first and last names, and to ensure that users select the proper recipient for an email.

### Table 733: Email client properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.ui.email_client.email_address.disambiguator</td>
<td>Sets the columns from the User [sys_user] table that the autocomplete list displays. Separate each column name with a semicolon character (;). See the system dictionary for a list of available column names. For example, to add the sys_user.email and sys_user.company columns, set the property to: email;company.</td>
</tr>
<tr>
<td></td>
<td>• Type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: name</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Property [sys_properties] table</td>
</tr>
</tbody>
</table>
Figure 785: Auto-complete in the email client

Control access to the email client
You can control access to the email client by changing an ACL rule.
Only users with the itil role can access the email client. The following ACL rule controls this access:

Table 734: ACL rule

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmailClientProcessor</td>
<td>processor</td>
<td>execute</td>
</tr>
</tbody>
</table>

Note: Because the visibility of the email icon is determined by whether the current user has write access to the table, it is possible that a user may be able to see the email icon and still not open the email client.

Create an email client template
You can create a different template for each table that uses the email client.
Role required: admin
The email client uses its own email templates to define default values for fields.
1. Navigate to System Notification > Email > Client Templates.
2. Click New.
3. Fill in the fields on the Email Client Template form.

Table 735: Email client template form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique template name.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Table</td>
<td>Select the table that the template applies to. You must enable the email client for the same table.</td>
</tr>
<tr>
<td>To</td>
<td>Enter a comma-separated list of either field names that contain user email addresses or specific email addresses.</td>
</tr>
<tr>
<td>Cc</td>
<td>Enter a comma-separated list of either field names that contain user email addresses or specific email addresses.</td>
</tr>
<tr>
<td>Bcc</td>
<td>Enter a comma-separated list of either field names that contain user email addresses or specific email addresses.</td>
</tr>
<tr>
<td>Subject</td>
<td>Enter a description of the email. The description can contain a comma-separated list of field names or specific values.</td>
</tr>
<tr>
<td>Body</td>
<td>Enter any text or mail script that you want to appear in the message body.</td>
</tr>
</tbody>
</table>

4. Click Submit.

**Note:** If the Client Templates module is not visible, enable the module.

Here is a sample template for the Incident table.

```javascript
template.print("Incident number -" + current.number + ":\n");
```

Here is how the template populates an incident email.
Set the from address with an email client template
Use an email client template to set a default value for the From field if it should be different from the system default.

Role required: admin

For example, you can dynamically set the From field to the email address of the current user.

1. Navigate to System Policy > Email > Client Templates.
2. Select an existing template or create one. For example, select the Incident Template record.
3. Configure the form to add the From field.
4. Enter a script to add a default value to the From field.

For example, the following script displays the current user name and the current user email address.

```
javascript:gs.getUserDisplayName() + " <" + gs.getUser().getEmail() + " >"
```

Display an editable From field
By default, the email client does not display a From field on email messages, but you can add one and have it be editable.

Role required: admin

By default, the email client uses an email address taken from the email account. To display an editable From field in the email client:

1. Navigate to System Properties > UI Properties.
2. Select the **Yes** check box for Override the email 'From:' address in the email client 
(glide.ui.email_client.from).

*Display the Reply To field*

By default, the email client does not display the Reply to field because users cannot change this address. But you can display this field using a UI property.

Role required: admin

The email client always uses the same Reply to address as that defined for email notifications. This ensures that the email has a valid *watermark* and can generate *inbound email actions* as normal. If you want to change the global reply-to address you must configure your instance to use your own SMTP server.

1. Navigate to **System Properties** > **UI Properties**.
2. In the Override the email 'Reply to:' address in the email client (glide.ui.email_client.reply_to) field, select the **Yes** check box.

*SMS delivery with the email client*

A property is available that lets the user select an option to send a notification via SMS.

The *Subscription Based Notifications plugin* activates the following *system property*:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.email_client.show_sms_option</td>
<td>Specify whether a check box appears in the email client for sending the message to the user's SMS device. If no SMS device exists, the email client sends the message to the primary email device.</td>
</tr>
</tbody>
</table>
Quick messages allow you to insert predefined text into the message body of the email client. Role required: admin.

Selecting a quick message fills the Message Text field with the body specified in the quick message. After you define one or more quick messages, the Quick Message selector appears in the email client. Use the following syntax: \${variable_name}.

**Note:** If you add attachments to the Quick Message templates, the attachments are not sent as part of the email distribution.

1. Navigate to **System Policy > Email > Quick Messages**.
2. Click **New**.
3. Fill in the Email Client Canned Messages form.

**Table 737: Email client canned messages form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Enter a name to appear in the Quick Message selector.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the quick message available for selection. Clear this check box to remove the message from the Quick Message selector.</td>
</tr>
<tr>
<td>User</td>
<td>Select the user who has access to this quick message. Selecting a user restricts access to that user only. Leave the field blank to have no user-based restrictions.</td>
</tr>
<tr>
<td>Group</td>
<td>Select the group whose members have access to this quick message. Selecting a group restricts access to members of that group only. Leave the field blank to have no group-based restrictions.</td>
</tr>
<tr>
<td>Body</td>
<td>Enter the text you want to insert into the Message Text field. By default, the field supports HTML format.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.
Email icon display
You can use access control rules to hide or display the email icon on forms.

Users with write access to the current table can see the email icon. To remove the icon, remove the user’s write access to the table. Typically you do this in one of two ways:

• Make the user an Employee Self-Service (ESS) user. ESS users do not have a user role, and without a role they do not have write access to the Incident table. Making a user an ESS users, therefore, hides the email icon on the Incident form.
• Create a custom ACL rule and user role that does not have write access to the table. The default ACL rule for the email client checks to see if the user has the itil role. If you grant users a custom role other than itil, then any such users will not see the email icon.
Push notifications

In addition to sending email and SMS notifications, an instance can send push notifications to mobile devices.

A push notification is a text message that appears on a user’s mobile device to alert them about something important or to ask them to perform an action. Starting with the Geneva release, your instance supports push notifications.

Figure 787: An example push notification

Use push notifications to send messages to users when certain conditions are triggered on your instance, such as the assignment of an incident to the user. A push notification can even ask for a reply, and the instance can process the reply by taking action on related records. For example, you can have the instance send an approval request for a Change to a user, and let the user approve or deny the Change by clicking a response button on push notification. The user's response can then update the status of the Change record.

You can set up push notifications in a similar manner to email and SMS notifications. Determine:

- who to send the notification to
By default, the ServiceNow mobile application supports push notifications. But you can also develop your own push application and configure your instance to send push notifications to it.

**Push notification setup**

Push notification setup differs depending on the mobile or push application that you want your users to use. For an overview, see:

- **Push notification setup with the ServiceNow mobile application** on page 2872

  **Note:** Push notifications for on-premise instances are not supported.

- **Push notification setup with a custom push application** on page 2872. If you create your own application, you must understand how push notifications and the Apple Push Notification Service system works. For more information, see the APNs Overview in the *Local and Remote Notification Programming Guide* for Apple developers.

**Attention:** Apple does not guarantee delivery of all push notifications. Review the Quality of Service (QoS) information in the *Local and Remote Notification Programming Guide* for Apple developers.

**The push notification system**

Several elements are involved in the push notification system.

- **Customer instance**: your instance.
- **Push proxy**: An instance that collects all push notifications that go to the ServiceNow mobile application and forwards them to the ServiceNow iOS application. If you create a custom push application, you do not use the push proxy.
- **Push provider**: The provider of push messages, which by default is the Apple Push Notification Service (APNs) for the ServiceNow mobile application.
- **Feedback provider**: The provider of feedback messages, which tell the instance what devices are no longer valid. By default, the Apple feedback server handles feedback messages for the ServiceNow mobile application.
- **Feedback proxy**: The ServiceNow instance that handles feedback messages from the Apple feedback server.
- **Push application on a mobile device**: The application, such as the ServiceNow mobile application. You can also build and customize your own push application.

**Push notification process**

The push notification process is as follows:

1. Activity on the instance triggers a push notification.
2. The instance looks for who to send the notification to and checks the user’s preferences to find their push device settings.
3. The instance sends the push notification to the push notification service, such as the Apple Push Notification Service (APNs). Optionally the instance can send the notification through a push proxy instance, which then forwards the notification to the APNs.
4. The push notification service sends the notification to the user’s push application. When the instance sends the notification to the push notification service, it includes message content along with the instance name and ID so that a response to the notification can be sent back to the correct instance.

5. If the user can reply to the notification, such as sending an approval to a change request, the response is sent via REST message back to the instance.

6. The instance identifies a script to run to handle the response.

7. The script takes action on the instance, such as marking a change request approved. If there is invalid JSON or if the script fails, an error response is sent back to the device.

This diagram shows the elements of the push notification system. Within the ServiceNow infrastructure are your instance, the optional push proxy instance, and the ServiceNow feedback proxy. External to ServiceNow are the APNs, and the user's mobile device, which includes the application that is registered to receive push notifications. See Push feedback on page 2872 for an explanation of the feedback servers.

Figure 788: The push notification system

Push notification responses

One of the benefits of push notifications is that users can take action on records in the instance by responding to a push notification.
The actions that they can take are defined in a script that you associate with the notification message. The response is sent to the instance in this format:

http://{instance_name}/api/now/v1/push/{application Name}/action/{action}

The application name is the push application the user is using. This must match the application name in the Push Application [sys_push_application] table. The action is the Sys ID of the action in the Push Notification Action Script [sys_push_notif_act_script] table.

Push feedback

*Feedback* refers to the information about failed message delivery and the push applications that can no longer receive push notifications.

The instance uses a scheduled job to pull feedback data every hour from the APNs. When the APNs determines that a device can no longer receive push messages, the corresponding device in the Push Installation [sys_push_installation] table is set to inactive, thereby making it unable to receive push notification messages. If the same device is again able to receive push notifications, the device receives the same token and a new record is inserted into the Push Notification Installation [sys_push_notif_app_install] table.

Push notification setup with the ServiceNow mobile application

If you are using the default ServiceNow mobile application, push notifications are set up similar to subscribable notifications.

**Note:** Push notifications with the ServiceNow mobile application are not supported in on-premise instances.

The process is as follows:

1. A user installs the ServiceNow mobile application and agrees to accept push notifications. The device then:
   - Obtains a token that identifies the device
   - Triggers the creation of a device in the user’s notifications preferences
   - Automatically subscribes the user to the push-specific notifications that are set up for the ServiceNow mobile app.

   When the instance receives the push notification acceptance message, it creates a record in the Push Notification Installation [sys_push_notif_app_install] table. This record is what the instance uses to identify the device + mobile application combination necessary to identify a push notification recipient.

2. The system administrator sets up push notifications, just like they set up an email notification. Push notifications contain two parts:
   - The push message
   - The notification, which includes the push message

3. Users or administrators can verify the user's notification preferences to see if the user has a push device configured and that the user is subscribed to the relevant push notifications.

Push notification setup with a custom push application

If you are using your own mobile or push application, push notifications there are a number of steps to take in addition to setting up push notifications with the ServiceNow mobile application.

The process is as follows:
1. The push app developer creates a record in the Push Application [sys_push_application] table for the custom application. This record is what the instances uses to identify the device + push application combination necessary to identify a push notification recipient.

2. The push app developer creates push notification action scripts to tell the instance how to handle responses to push notifications.

3. The push app developer can also create a content payload in JSON for different types of push notifications. The content determines how a push notification appears on the push application, and whether or not the user can send a message in response to the push notification.

4. The system administrator sets up push notifications, and specifies the custom push app and, if desired, the content.

5. Users set up their push device on their user profile's notification preferences, just as they can set up email addresses or devices for SMS messages. Administrators can also set up these preferences on behalf of users.

Activate push notifications

Several plugins must be activated to use push notifications. If you have the Mobile UI (com.snc.service_management_m) plugin active, push notification plugins are automatically activated.

Role required: admin

You must have the following plugins active on your instance:

- **Push Notification**: Provides the necessary components and REST APIs to send push notifications to mobile devices.
- **Notification System Push Addon**: Adds support for push notifications to the existing notification system.

This plugin is installed in the push notification infrastructure:

- **Push Feedback**: Handles feedback from Apple on which devices are no longer valid so they do not keep receiving push notifications.

1. In the HI Service Portal, click **Service Catalog > Activate Plugin**.
2. Fill out the form.

<table>
<thead>
<tr>
<th>Target Instance</th>
<th>Instance on which to activate the plugin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin Name</td>
<td>Name of the plugin to activate.</td>
</tr>
<tr>
<td>Specify the date and time you would like this plugin to be enabled</td>
<td>Date and time must be at least 2 business days from the current time.</td>
</tr>
</tbody>
</table>

**Note**: Plugins are activated in two batches each business day in the Pacific timezone, once in the morning and once in the evening. If the plugin must be activated at a specific time, enter the request in the Reason/Comments.
3. Click Submit.

Installed with push notifications

Several types of components are installed with the push notifications plugins.

Tables installed with push notifications

These tables are installed with push notifications.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push Application [sys_push_application]</td>
<td>Push applications registered to the instance to receive push messages.</td>
</tr>
<tr>
<td>Push Default Registration [sys_push_notif_default_reg]</td>
<td>Contains all of the notifications users are automatically subscribed to for a given push application. You can add notifications to this table on the Push Default Registrations related list on the Push Application form.</td>
</tr>
<tr>
<td>Push Feedback [sys_push_feedback]</td>
<td>Feedback from the APNs that tells the instance which push devices can no longer receive push messages. The feedback is handled either on the same instance where your notifications are triggered, or a central instance that uses a REST call to tell your instance which push applications to deactivate.</td>
</tr>
<tr>
<td>Push Message Attribute Definition [sys_push_notif_msg_attr_def]</td>
<td>The attribute definitions used for push message content specification.</td>
</tr>
<tr>
<td>Push Message Attribute Value [sys_push_notif_msg_attr_val]</td>
<td>The values associated with push messages.</td>
</tr>
<tr>
<td>Push Notification [sys_push_notification]</td>
<td>The push notifications that the instance attempted to send to users.</td>
</tr>
<tr>
<td>Push Notification Action Script [sys_push_notif_act_script]</td>
<td>The scripts that the instances uses in response to an actionable push message.</td>
</tr>
</tbody>
</table>
### Push Notification Installation

**Table name**: [sys_pushnotif_app_install]  
**Description**: The devices with push apps where users agreed to receive push notifications. The records in this table are listed by the token for the push app. The instance uses this information to know which push device and app to send notifications to. Records are created in this table when a user logs on an instance with the push app. The Mobile Devices [sys_mobile_devices] table, which is installed with the Mobile UI plugin, contains all the user devices that logged into the instance with the ServiceNow mobile application.

### Push Notification Message

**Table name**: [sys_pushnotif_msg]  
**Description**: Messages customized for push notifications. These messages can be associated with a notification.

### Push Notification Message Content

**Table name**: [sys_pushnotif_msg_content]  
**Description**: The entire content, including JSON, for push messages.

### Push Platform

**Table name**: [sys_pushplatform]  
**Description**: The platforms that are supported for push notifications, and the maximum payload size.

#### Business rules installed with push notifications

These business rules are installed with push notifications.

<table>
<thead>
<tr>
<th>Business rule</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Device and Subscriptions</td>
<td>Push Installation [sys_pushnotif_install]</td>
<td>Automatically creates a notification device on a user's notification preferences when the user registers a mobile application with the instance.</td>
</tr>
</tbody>
</table>

#### Outbound REST messages installed with push notifications

These outbound REST messages are installed with push notifications.

<table>
<thead>
<tr>
<th>REST message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceNowMobileApp Push</td>
<td>The REST message that you can use with your custom iOS mobile app.</td>
</tr>
</tbody>
</table>

#### Roles installed with push notifications

These roles are installed with push notifications.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>push_admin</td>
<td>Can create and modify push notifications.</td>
</tr>
</tbody>
</table>

#### Push components installed with push notifications

These components are installed with push notifications.
Table 738: Push applications installed with push notifications

<table>
<thead>
<tr>
<th>Push application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceNowPushApp</td>
<td>The push application record for the ServiceNow mobile application.</td>
</tr>
</tbody>
</table>

**Note:** The ServiceNow mobile application automatically subscribes users to several push notifications. You can see the list of these notifications in the Push Default Registrations related list on the ServiceNowPushApp application record.

Table 739: Message content installed with push notifications

<table>
<thead>
<tr>
<th>Message content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectMessageContent</td>
<td>The payload for ServiceNow connect messages.</td>
</tr>
<tr>
<td>Generic Approval (Background)</td>
<td>The payload that provides an approval and rejection option to the user receiving the push message.</td>
</tr>
<tr>
<td>Generic Approval (Foreground)</td>
<td>The payload that provides an approval and rejection option to the user receiving the push message.</td>
</tr>
<tr>
<td>Generic Record Payload</td>
<td>A generic payload that you can use to send push messages to users.</td>
</tr>
</tbody>
</table>

Table 740: Push notification actions installed with push notifications

<table>
<thead>
<tr>
<th>Push notification action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval - Approve</td>
<td>Allows an administrator to approve a request.</td>
</tr>
<tr>
<td>Approval - Reject</td>
<td>Allows an administrator to reject a request.</td>
</tr>
</tbody>
</table>

Table 741: Push notifications installed with push notifications

<table>
<thead>
<tr>
<th>Push notifications</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectMessagePushNotification</td>
<td>Live Feed Message [live_message]</td>
<td>Sends a push notification when a new live feed message</td>
</tr>
</tbody>
</table>

Set up push notifications for the ServiceNow mobile app

Administrators can set up push notifications by creating push-specific notifications and associating them with standard platform notifications.

The **Push notification plugin** must be active.
Role required: admin

Follow these steps to set up push notifications that send content to users.

Create a push message

Before you create a push notification, create a push message with the actual message content.

Role required: admin

2. Fill out the fields on the form (see table).
3. Click Submit.
Table 742: Push Notification Message form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for the message.</td>
</tr>
<tr>
<td>Push app</td>
<td>Select a push app that this message can be sent to.</td>
</tr>
<tr>
<td>Push Message Content</td>
<td>Select content layout for the message.</td>
</tr>
<tr>
<td>Message</td>
<td>Enter the message. You can add variables just as you would for other notifications. Any message you enter here overrides the message in the notification.</td>
</tr>
<tr>
<td>Related list</td>
<td></td>
</tr>
<tr>
<td>Push Message Attribute Values</td>
<td>Select the attributes that apply to this notification. The attributes on this form override those associated with the content layout.</td>
</tr>
</tbody>
</table>

Make sure that the relevant users have their push devices configured and that the users subscribe to the notification in the user's notification preferences. See Set up a mobile notification device for a user on page 2895, Select notifications on page 2898, and Subscribe to a notification message on page 2878.

Create a notification using a push message

Email administrators can create a notification that specifically sends a push notification.

Configure the push message and create a push message content before performing these steps.

Role required: admin

You can associate a push message with a standard notification.

1. Navigate to System Notification > Create Push Notification.
2. Fill out the notification form as necessary (see Create an email notification on page 2756 for descriptions of the form fields.
3. Click the What it will contain tab.
4. Next to Push Messages, click the lock icon and select a push message.
5. If you want this notification to be sent only as a push notification and not as any other type of notification, select Push Message Only.
6. Click Submit.

If the notification fails, the user is not notified. If the message fails to send because it exceeds the maximum payload, the instance logs the failure in the System Log.

Make sure that the relevant users have their push devices configured and that the users subscribe to the notification in the user's notification preferences. See Set up a mobile notification device for a user on page 2895, Select notifications on page 2898, and Subscribe to a notification message on page 2878.

Subscribe to a notification message

After setting up your devices, you are ready to subscribe to new messages or manage messages you have already received.
If you have subscribed to messages, your list of notification messages can build over time. You can create preferences for how and when these messages are delivered, or unsubscribe to messages that are not configured as mandatory.

**Note:** Filters that you apply to messages do not override the original filters that the administrator created for the notifications. Your filter conditions are evaluated after the original conditions are met. If the administrator's notification filter fails, your filter conditions are not evaluated. If neither an advanced filter nor a predefined filter is selected, you see all notifications that match the administrator's original filter conditions.

1. Navigate to **Self-Service > My Profile** to open your user profile.
2. Click the **Notification Preferences** related link.
3. On the Notification Preferences page, click the plus icon.
4. Fill in the fields as described in the table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification Message</td>
<td>Select a message from the list. Related fields appear for certain notifications. For example, if you select CI affected or Location affected, a field appears for selecting the CI or the location. Duplicate messages are not permitted on a device.</td>
</tr>
<tr>
<td>Device</td>
<td>Select the notification device for the message. The device that you are adding a notification for is selected by default, but you can change the device, if appropriate.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Select a schedule for message delivery.</td>
</tr>
<tr>
<td>Filter</td>
<td>Select a notification filter to refine the notifications that you receive. For example, you might select a filter whose conditions send notifications when an incident with a priority of 1 - Critical is opened for a network issue. The system evaluates the conditions in this filter after the conditions in the administrator's notification filter. This field is not available when an advanced filter is configured.</td>
</tr>
<tr>
<td>Advanced filter</td>
<td>Select the check box to enable the condition builder and select the table to use. When you select the check box, the Filter field is replaced by the Table and Conditions fields.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the table for the notification. For example, if you select the CI affected notification message, you might select the Incident [incident] or Change Request [change_request] table. This field appears when you check the Advanced filter check box.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Define as many conditions as needed to refine the notification.</td>
</tr>
</tbody>
</table>
If you do not specify a schedule or filter, the message is delivered when the event occurs and when the filter condition defined by the administrator is met. You can select a delivery schedule that is more convenient, create conditions that refine the notifications you receive, or select a predefined filter intended for reuse. For example, the default filter for a notification might send you a message when a high priority incident is opened for a particular CI. You can refine the message delivery options and add a filter that notifies you when those incidents are assigned to a particular assignment group.

5. Click **Save**.

---

**Set up push notifications for a custom push app**

Mobile application developers can set up a customized push app and configure components such as action scripts and notification layouts.

The **Push notification plugin** must be active.

Role required: admin or push_admin

Follow these steps to set up a push notification infrastructure using your own push application.

---

**Caution:** General system administrators do not need to configure these settings to use the ServiceNow mobile application with push messages. These instructions are intended for users who develop their own customized push application.

---

**Create a push application**

You must register your customized mobile application with your instance so it can receive push notifications.

Role required: admin or push_admin
Push notifications are application specific: they are sent to one type of mobile application, regardless of how many users have this application installed. By default, the ServiceNow mobile application is automatically set up and ready to use starting with the Geneva release. If you develop your own mobile application, you must create a record for it in the Push Application table.

1. Navigate to **System Notification > Email > Push Application**.
2. Fill out the fields on the form (see table).
3. Click **Submit**.
A push application is a mobile app that can receive push notifications from a ServiceNow instance. Create a mobile application record for your customized mobile app and specify how to handle push messages and feedback.

### Push Application Details

**Name:** ServiceNowPushApp  
**Application:** Global  
**Push:** REST API

### Push Notification Message Contents

<table>
<thead>
<tr>
<th>Message Content</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>var json = {</td>
</tr>
<tr>
<td></td>
<td>&quot;aps&quot;: [</td>
</tr>
</tbody>
</table>
|                 |   "sound": "def...

Push app = ServiceNowPushApp
### Table 743: Push Application form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for the layout.</td>
</tr>
<tr>
<td>Push</td>
<td>Select from the following:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Direct</strong>: Send push notifications directly to the push service without going through an intermediary. This is the recommended method. If you select this option, you need to specify a X.509 <strong>Certificate</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>REST API</strong>: Send push notifications through an intermediary instance, that in turn sends the notification to the APNs. The commutation between the two instances is via a REST message.</td>
</tr>
<tr>
<td></td>
<td>• <strong>None</strong>: do not enable push notifications for this application.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Select from the following:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Apple</strong>: Send push notifications directly to the APNs without going through an intermediary.</td>
</tr>
<tr>
<td></td>
<td>• <strong>REST API</strong>: Send push notifications through an intermediary instance, that in turn sends the notification to the APNs. The commutation between the two instances is via a REST message.</td>
</tr>
<tr>
<td></td>
<td>• <strong>None</strong>: Do not handle feedback for this push application.</td>
</tr>
<tr>
<td>Apple</td>
<td>The X.509 <strong>Certificate</strong> and <strong>Sandbox Certificate</strong> you created on the Apple notifications portal. The certificate allows a device to talk to the APNs. This option appears only if you select <strong>Direct</strong> for the <strong>Push</strong> field.</td>
</tr>
<tr>
<td>Google</td>
<td>Select the API key you obtain from Google for the android push notification.</td>
</tr>
<tr>
<td>Related lists</td>
<td><strong>Push Notification Message Contents</strong> Select the content layout associated with this app.</td>
</tr>
<tr>
<td></td>
<td><strong>Push Default Registrations</strong> Select the notifications that you want automatically subscribed to users who use this application. Users are subscribed to only active notifications.</td>
</tr>
</tbody>
</table>
Create a push action

A push action is a server-side script that runs when the instance receives a response to an actionable push message.

Create a push action to perform an action on the instance. For example, you might have an actionable push message that lets the user approve a change request. The action that handles the response should update the **Approval** field on the relevant Change Request record.

You can use global variables or, optionally, current variables and parameters passed through the **JSON content**.

Role required: admin or push_admin

1. Navigate to **System Notification > Push > Push Action**.
2. Fill in the form fields (see table).
3. Click **Submit**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for the action.</td>
</tr>
<tr>
<td>Script</td>
<td>Enter the script.</td>
</tr>
</tbody>
</table>

See *Push notification example* on page 2889 for an example of a push action.

Create push message content

Push message content specifies additional JSON content in the push notification payload that is sent to the push provider.

You must know how to use JSON with push messages.

Role required: admin or push_admin

The content layout defines the style of push notification that can be sent out. Push contents give you the opportunity to add content, such as a picture, as well as provide action buttons or icons. Use these in the script:

- **current**: the properties of the current record.
- **message**: the push message sent as the body of the entire push content.
- **attributes**: the object of the push message *attributes that you define*.

1. Navigate to **System Notification > Push > Push Message Content**.
2. Fill out the fields on the form (see table).
3. Click **Submit**.
Push notification message content specifies additional JSON content in the push notification payload that is sent to the push provider. Use this record to create the JSON content and specify the provider-specific values, such as a sound or badge.

The following variables are available for use in the script:

- `current`: access properties of the record that triggered this push notification, such as the fields on an incident record.
- `message`: access contents of the message that will be sent as the body of the push notification.
- `attributes`: access attributes defined for this push notification.

```javascript
var json = {
  "app": "default",
  "record": {
    "table": current.getTableName(),
    "sys_id": current.sys_id
  }
};
```
Table 745: Notification Message form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for the layout.</td>
</tr>
<tr>
<td>Push app</td>
<td>The push application the content can be used with.</td>
</tr>
<tr>
<td>Push Message Generation</td>
<td>Enter a script that determines the layout.</td>
</tr>
<tr>
<td></td>
<td>See the example scripts below.</td>
</tr>
<tr>
<td>Related list</td>
<td></td>
</tr>
<tr>
<td>Push Message Attribute Definitions</td>
<td>Select the attributes that apply to this notification. Attributes can be a value or an action. These attributes are used as default values for the content items you create in the content script. However, any attributes you create with the push message can override these attributes. See Create an attribute definition on page 2886 for information on creating attributes.</td>
</tr>
</tbody>
</table>

The following is an example of a content record that creates a two-button layout, one to approve something, such as a change request, and one to decline it.

```javascript
var json = {
  "table" : current.getTableName(),
  "sys_id" : current.sys_id,
  "template" : {
    "type" : "2 button",
    "button1" : {
      "title" : "Approve",
      "action" : attributes.button_action,
      "parameters" : {
        "response" : "approve"
      }
    },
    "button2" : {
      "title" : "Decline",
      "action" : attributes.button_action,
      "parameters" : { "response" : "decline"
    }
  }
};
json;
```

Create an attribute definition

Push message attribute definitions allow you to create reusable properties for push message content specification.

Role required: admin or push_admin
Use an attribute definition to specify a default push action script or string that you can then use in the push message content.

These attributes can be overridden by any *push message attribute values* that you create.

1. Navigate to **System Notification** > **Push** > **Push Message Content**.
2. Select a content layout record or script record.
3. In the Push Message Attribute Definition related list, click **New**.
4. Fill out the fields on the form (see table).
5. Click **Submit**.

**Table 746: Push Message Attribute Definition form fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Name</td>
<td>Enter a descriptive name for the layout.</td>
</tr>
</tbody>
</table>
Create an attribute value or action for a push message

Attribute values are associated with push messages to provide a way for the message to include certain types of information in the push notification.

Create default definitions on the Push Message Attribute Definitions related list of the Push Message Content form.

Role required: admin or push_admin

These push message attribute values override values that you set in an attribute definition on the Push Content form.

2. Select a message.
3. In the Push Message Attribute Values related list, click New.
4. Fill out the fields on the form (see table).
5. Click Submit.

See Push notification example on page 2889 for an example of a push attributes.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Type</td>
<td>Select the type of attribute:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Action</strong>: An action to take on the instance, as defined by a script.</td>
</tr>
<tr>
<td></td>
<td>• <strong>String</strong>: An arbitrary string to send as part of the message content.</td>
</tr>
<tr>
<td></td>
<td>The string can specify items like a button label in the message.</td>
</tr>
<tr>
<td>Default Script</td>
<td>Select the mobile action script that tells the instance what to do when it</td>
</tr>
<tr>
<td></td>
<td>receives a response from the push notification. This option appears if you</td>
</tr>
<tr>
<td></td>
<td>select <strong>Action</strong> for the <strong>Type</strong>.</td>
</tr>
<tr>
<td>Default Value</td>
<td>Specify an arbitrary string value to be placed in the attribute that is used</td>
</tr>
<tr>
<td></td>
<td>by the message content. For example, the string could specify a button</td>
</tr>
<tr>
<td></td>
<td>label. This option appears if you select <strong>String</strong> for the <strong>Type</strong>.</td>
</tr>
</tbody>
</table>
### Push notification example

This example illustrates a customized push notification, including customized push actions and content. In this example, you will learn how to create a push notification for incidents that are not assigned to anyone. This example assumes you created your own custom push application that is referred to in this example as **MyPushApp**. The notification will be sent to a user who can respond by accepting assignment to the incident or ignoring it.

#### The push actions

First you will create push actions that give the user options on how to respond to the push notification. In this example, create two actions: **assign to me** and **ignore**. The content of the actions is as follows:
• **Assign to me:**

```javascript
(function runAction(/*Optional GlideRecord*/ current) {
    if (current.getTableName() == "incident" && current.assigned_to == undefined) {
        current.comments = "I have self assigned this incident to myself";
        current.assigned_to = gs.getUser();
        current.update();
    }
})(current);
```

• **Ignore:**

```javascript
(function runAction(/*Optional GlideRecord*/ current) {
    gs.log(gs.getUser() + " ignored this record: " + current.number);
    //writes a message to the system log that the user ignored this incident.
})(current);
```

See *Create a push action* on page 2884 for specific steps on how to create these actions.

### The push content

Next, you will create the JSON content that specifies two buttons that the user will use to respond to the incident. The content is as follows:

```javascript
(function buildJSON(/*GlideRecord*/ current, /*String*/ message, /*Object*/ attributes) {
    var json = {};

    json = {
        "aps": {
            "sound": "default"
        },
        "record": {
            "table": current.getTableName(),
            "sys_id": current.sys_id
        },
        "action": {
            "button1_text": attributes.name_of_button1,
            "button1_action": attributes.action_of_button1,
            "button2_text": attributes.name_of_button2,
            "button2_action": attributes.action_of_button2
        }
    );
    return json;
})(current, message, attributes);
```

The content script specifies two labels and two actions, one for each button. You must create definitions for these. But later you will override these definitions with the actual button labels and actions on the push message. Here are the definitions:
The push message

Next you will create a push message that specifies the message that appears to the user: **Can you take on a new incident?**. The push message associates your custom application **MyPushApp** with the content you just created.
Figure 790: Push message example

The Push Message Attribute Values related list specifies the four items needed for the button label and the action the instance takes when the user clicks that button:
### Table 748: Push message attribute values

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action_on_button2</td>
<td>Ignore</td>
<td>Ignore</td>
<td>The Ignore action is what you specified as a push action. This is associated with button 2.</td>
</tr>
<tr>
<td>action_on_button1</td>
<td>Assign to me</td>
<td>Assign to me</td>
<td>The Assign to me action is what you specified as a push action. This is associated with button 1.</td>
</tr>
<tr>
<td>name_of_button2</td>
<td>Ignore</td>
<td>none</td>
<td><em>Ignore</em> is the label the user sees for button 2.</td>
</tr>
<tr>
<td>name_of_button1</td>
<td>Assign to me</td>
<td>none</td>
<td><em>Assign to me</em> is the label the user sees for button 1.</td>
</tr>
</tbody>
</table>

**The push notification**

Finally, you will create a notification that is triggered when a new incident is unassigned. It is sent to a group named Incident Manager. You will then associate the push message that you just created with it. The notification is as follows:
Figure 791: Push notification example
Set up a mobile notification device for a user

When a user registers their mobile application to an instance, a notification device is automatically created for them in their notification preferences. You can still set up or modify mobile notification devices for your users.

If your users have the ServiceNow mobile application installed on their devices, you do not need to follow these instructions.

1. Navigate to User Administration > Users.
2. Select a user.
3. Click Notification Preferences.
4. Click New Device.
5. In the Type field, select Push.
6. In the Push app field, select the application that is already configured to handle push notification for the instance.

See Set up a notification device on page 2896 for more information on the other fields in this form.

You can also disable the push device for the user if the user should no longer receive push notifications.

**Note:** You do not need to disable the push device for the users who remove the push app or who no longer use their push device. The instance automatically deactivates their device in the Push Notification Installation table.

7. Subscribe to the push notifications or instruct the user to subscribe in their notification settings.

Failed push notification messages

Push notification delivery might fail for a number of reasons. You can view which messages failed and re-queue them to be sent out if necessary.

Role required: admin

**Note:** There is no way for the instance to guarantee or confirm push message delivery. See the [iOS developer library](https://developer.apple.com/library/mac/documentation/NetworkingInternetTrading/Conceptual/RemoteNotificationsPG/) for more information about how Apple handles push notifications.

Messages can also fail to send for several reasons, such as a large message queue or other issue with the instance. You can use a system property to control how long the instance queues a push notification after it is triggered.

Push notification message payload size is limited by provider:

- Apple iOS: **2,048 bytes**
- Google: **4,096 bytes**

Your instance does not send push messages that exceed this limitation. System logs save any failed messages. See the [iOS Developer Library payload documentation](https://developer.apple.com/library/mac/documentation/NetworkingInternetTrading/Conceptual/RemoteNotificationsPG/Payloads.html) for more information on Apple payload limitations.

1. Navigate to System Logs > Push Notifications.

   By default, the messages that were created today appear. You can change the filter if necessary. The Type column can have these values:

   - **failure**: The message could not be sent.
   - **pending**: The message is queued and will be processed.
• **success**: The message was successfully sent, although not necessarily received by the mobile device.

2. Put any failed messages back into the queue to be resent by selecting the check boxes next to the failed messages, and then selecting **Re-queue failed push notifications** from the choice list.

3. If a number of push notifications continue to fail, consider increasing the value in the glide.push.notification.ttl_seconds property. See **Push notification properties** on page 2896 for more information.

### Push notification properties

Push notifications provides several properties to customize the setup.

Add these properties to the System Properties [sys_properties] table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.push.debug</td>
<td>Creates entries in the system log for push notification errors.</td>
<td>false</td>
</tr>
<tr>
<td>glide.push.feedback.debug</td>
<td>Creates entries in the system log for feedback sent by the APNs.</td>
<td>false</td>
</tr>
<tr>
<td>glide.push.notification.ttl_seconds</td>
<td>Specifies a time period after which a queued push notification expires. After a push notification is triggered, it is queued and finally processed by a scheduled job. If the time the notification sits in the queue exceeds this value, the notification is not sent. Check the <strong>Push Notifications log</strong> for more information.</td>
<td>21600 seconds (6 hours)</td>
</tr>
</tbody>
</table>

### Subscription-based notifications

Subscription-based notifications enable users to proactively subscribe to items that interest them and unsubscribe from messages that are not mandatory.

Users can also specify additional notification devices, such as mobile phones, that each of their notifications can be configured to use.

Before users can manage the notifications that are sent to them, administrators must create email notifications to which users can subscribe. Administrators can also make subscription-based notifications mandatory so users cannot unsubscribe to them. Then users can subscribe or unsubscribe to the notifications, and add schedules and filters to the subscription to limit the notifications that can be received.

### Set up a notification device

You can add devices for a user to receive notifications. A **device** is a mobile device, email account, or voice message system that the user has access to.
Role required: admin

Notification devices include email addresses, service providers for SMS messages, and mobile applications.

1. Navigate to User Administration > Users.
2. Select a user.
3. Click Notification Preferences.
4. Click New Device.
5. Fill in the fields on the form (see table).
6. Click Submit.

---

**Table 750: New Device form**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A descriptive name of the device or account.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of device:</td>
</tr>
<tr>
<td></td>
<td>• Email: for email messages.</td>
</tr>
<tr>
<td></td>
<td>• SMS: for SMS messages.</td>
</tr>
<tr>
<td></td>
<td>• Mobile: for push notifications.</td>
</tr>
<tr>
<td>Email address</td>
<td>The user’s email address.</td>
</tr>
<tr>
<td>Mobile application</td>
<td>The <em>mobile application</em> for push notifications.</td>
</tr>
</tbody>
</table>
## Select notifications

Administrators can configure notification preferences for each user and users can also select which notifications they want to receive for various devices.

1. Navigate to **Self-Service > My Profile** to select your own notifications, or if you have admin access, navigate to **User > Administration Users** to select notifications for another user.
2. Click a user record to open it.
3. On the User form, click **Notification Preferences** under Related Links.

The notification preferences page appears. The preferences pages shows all the notifications available to the user and the devices that the user has configured, such as email, mobile phone, mobile device for push notifications. If a user does not have read-only access to the table on which the notification is based, the notification does not appear.

Every user has the **Primary email** device, where subscriptions to email notifications can be enabled.
4. Click the switch next to the notification to subscribe the user to it.

Modify notifications for a specific user device

After you set up the devices through which users can receive notifications, you can assign the notifications to each device and add advanced conditions to limit what notifications you can receive.

Role required: admin

1. On the Notification Preferences page for a user, find the notification from the list of notifications.
2. Click Edit next to the notification. The preferences for that notification appear.
3. Fill in the fields on the form (see table).
4. Click Submit.
Notification Preferences

Get notified whenever actions that involve you occur. Leverage filters or sort the list to focus on them, or unsubscribe entirely from messages. You can also create new notifications.

**Primary email**

- Change assigned to me
- Change assigned to my group
- Email assigned to group (sc_task)
- Problem assigned to me

**Admin mobile phone**

No notifications set up, click the plus button to add.

Figure 792: Notification preferences for a user's device

---

**Table 751: Form fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification Message</td>
<td>Select the notification. Related fields appear for certain notifications. For example, if you select <strong>CI affected</strong> or <strong>Location affected</strong>, a field appears for selecting the CI or the location. Duplicate messages are not permitted on a device.</td>
</tr>
<tr>
<td>Device</td>
<td>Modify the device if necessary.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Select a schedule that determines when the notification can and cannot be received.</td>
</tr>
</tbody>
</table>
### Field Description

**Filter**
Select a pre-configured filter with the criteria that determines when the notification can be sent. For example, you might select a filter whose conditions send notifications when an incident with a priority of 1 - Critical is opened for a network issue. The system evaluates the conditions in this filter after the conditions in the administrator's notification filter. This field is not available when an advanced filter is configured.

**Advanced filter**
Select this option if you want to use the condition builder to create additional criteria. When you select the check box, the Filter field is replaced by the Table and Conditions fields.

**Table**
Select the table for the notification. For example, if you select the CI affected notification message, you might select the Incident [incident] or Change Request [change_request] table. This field appears when you check the Advanced filter check box.

**Conditions**
Define as many conditions as needed to limit the notifications you receive.

### Create a service provider
Administrators can configure service providers for devices that use SMS.

**Role required:** admin

Administrators also have the option of configuring how a device’s service provider affects the construction of the device’s email address.

1. Navigate to the Notification Service Provider [cmn_notif_service_provider] table by typing cmn_notif_service_provider.list in the application navigator filter.
2. Fill out or change the fields on the form (see table).
3. Click **Update**.
4. Click **Save** on the Notification Preferences page.
### Table 752: The Notification Service Provider form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Configured name of the service provider.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of device, in this case <strong>SMS</strong>.</td>
</tr>
<tr>
<td>Active</td>
<td>Enables or disables this notification device.</td>
</tr>
<tr>
<td>Advanced notification</td>
<td>Removes all the previous options and displays the <em>Advanced script</em> field.</td>
</tr>
</tbody>
</table>

Abel Tuter’s Abel’s SMS device

![ServiceNow Notification Service Provider form](image-url)
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced script</td>
<td>Used for custom notifications that run a script rather than construct a traditional SMS/email. This is generally used when all outbound SMS messages must run through a central SMS hub, as opposed to being sent directly from the instance to the SMS provider. Advanced users can construct a script in this field that will send a notification to an old style numeric pager that cannot receive SMS communications. This field is rarely used and employs advanced scripts. Contact your representative to assist you with any advanced SMS scripting.</td>
</tr>
<tr>
<td>Construct address manually</td>
<td>Removes the prefix and suffix options and displays the <strong>Construction script</strong> field.</td>
</tr>
<tr>
<td>Construction script</td>
<td>Allows you to script the email address construction as you would in a business rule. For example, <code>abc + current.phone_number + def@text.att.net</code> would construct an email address of <code>abc2223334444def@text.att.net</code>. <strong>NOTE:</strong> <code>Current</code> is a reference to the device, not the service provider, hence the <code>current.phone_number</code> variable that uses the device's <code>phone_number</code> value.</td>
</tr>
<tr>
<td>SMS Provider Email Prefix</td>
<td>Places the provided text before the device's specified phone number; for example: <code>PREFIX2223334444@text.att.net</code></td>
</tr>
<tr>
<td>SMS Provider Email Suffix</td>
<td>Places the provided text after <code>@</code> sign; for example: <code>2223334444@SUFFIX</code></td>
</tr>
<tr>
<td>Notification Device Variables</td>
<td>Additional, optional attributes of an SMS device used inside an SMS service provider's <strong>Advanced script</strong>. This is generally used to deliver SMS notifications to an internal SMS distribution technology. This is an advanced scripting procedure and is not necessary for configuring external SMS providers.</td>
</tr>
</tbody>
</table>

**Select a service provider**

You can configure how a device's service provider affects the construction of the device's email address.

Role required: admin

1. Navigate to **User Administration** > **Users** and open any user's record.
2. Under Related Links, click **Notification Preferences**, and then click an SMS notification device.
3. If no SMS device is present, click **New Device** and configure one.
4. Select the appropriate service provider, and then click the reference icon for the Service provider field. The service providers are saved in the Notification Service Provider [cmnnotif_service_provider] table. Only active providers are visible.

The Notification Service Provider form provides the following fields:

Table 753: Selecting a service provider

<table>
<thead>
<tr>
<th>Field</th>
<th>Input value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Configured name of the service provider.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of device - in this case <strong>SMS</strong>.</td>
</tr>
<tr>
<td>Active</td>
<td>Enables or disables this notification device.</td>
</tr>
<tr>
<td>Advanced notification</td>
<td>Removes all the previous options and displays the Advanced script field.</td>
</tr>
<tr>
<td>Advanced script</td>
<td>Used for custom notifications that run a script rather than construct a traditional SMS/email. This is generally used when all outbound SMS messages must run through a central SMS hub, as opposed to being sent directly from the instance to the SMS provider. Advanced users can construct a script in this field that will send a notification to an old style numeric pager that cannot receive SMS communications. This field is rarely used and employs advanced scripts. Contact your representative to assist you with any advanced SMS scripting.</td>
</tr>
<tr>
<td>Construct address manually</td>
<td>Removes the prefix and suffix options and displays the Construction script field</td>
</tr>
</tbody>
</table>
Field | Input value
--- | ---
Construction script | Allows you to script the email address construction as you would in a business rule. For example, abc + current.phone_number + def@text.att.net would construct an email address of abc2223334444def@text.att.net.

*Note:* Current is a reference to the device, not the service provider, hence the `current.phone_number` variable that uses the device’s phone_number value.

SMS Provider Email Prefix | Places the provided text before the device's specified phone number (e.g. PREFIX2223334444@text.att.net).

SMS Provider Email suffix | Places the provided text after ‘@’ sign (e.g. 2223334444@SUFFIX)

Notification Device Variables | Additional, optional attributes of an SMS device used inside an SMS service provider's Advanced script. This is generally used to deliver SMS notifications to an internal SMS distribution technology. This is an advanced scripting procedure and is not necessary for configuring external SMS providers.

---

**SMS notification advanced scripting**

You can use these objects and their attributes in your advanced notification scripts on the SMS Notification Service Provider form.

**Table 754: Objects and attributes**

<table>
<thead>
<tr>
<th>Object</th>
<th>Example attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>current</td>
<td>• current.number</td>
</tr>
<tr>
<td></td>
<td>• current.assigned_to</td>
</tr>
<tr>
<td></td>
<td>• current.company</td>
</tr>
<tr>
<td></td>
<td>• current.state</td>
</tr>
<tr>
<td>Object</td>
<td>Example attributes</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>email</td>
<td>• email.notify</td>
</tr>
<tr>
<td></td>
<td>• email.sourceTable</td>
</tr>
<tr>
<td></td>
<td>• email.contentType</td>
</tr>
<tr>
<td></td>
<td>• email.attachments</td>
</tr>
<tr>
<td></td>
<td>• email.HTML</td>
</tr>
<tr>
<td></td>
<td>• email.dataVersionHeader</td>
</tr>
<tr>
<td></td>
<td>• email.UID</td>
</tr>
<tr>
<td></td>
<td>• email.allowSavingNoRecipientEmail</td>
</tr>
<tr>
<td></td>
<td>• email.recipients</td>
</tr>
<tr>
<td></td>
<td>• email.ignore</td>
</tr>
<tr>
<td></td>
<td>• email.save</td>
</tr>
<tr>
<td></td>
<td>• email.headers</td>
</tr>
<tr>
<td></td>
<td>• email.sysID</td>
</tr>
<tr>
<td></td>
<td>• email.attachmentLimits</td>
</tr>
<tr>
<td></td>
<td>• email.class</td>
</tr>
<tr>
<td></td>
<td>• email.sysId</td>
</tr>
<tr>
<td></td>
<td>• email.textBody</td>
</tr>
<tr>
<td></td>
<td>• email.hashCode</td>
</tr>
<tr>
<td></td>
<td>• email.weight</td>
</tr>
<tr>
<td></td>
<td>• email.equals</td>
</tr>
<tr>
<td></td>
<td>• email.logEmail</td>
</tr>
<tr>
<td></td>
<td>• email.reset</td>
</tr>
<tr>
<td></td>
<td>• email.wait</td>
</tr>
<tr>
<td></td>
<td>• email.body</td>
</tr>
<tr>
<td></td>
<td>• email.SMSText</td>
</tr>
<tr>
<td></td>
<td>• email.watermark</td>
</tr>
<tr>
<td></td>
<td>• email.textBodyLegacy</td>
</tr>
<tr>
<td></td>
<td>• email.sourceHeader</td>
</tr>
<tr>
<td></td>
<td>• email.subject</td>
</tr>
<tr>
<td></td>
<td>• email.instance</td>
</tr>
<tr>
<td></td>
<td>• email.importance</td>
</tr>
<tr>
<td>Object</td>
<td>Example attributes</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------</td>
</tr>
</tbody>
</table>
| device   | • device.service_provider  
|          | • device.group       
|          | • device.order       
|          | • device.sys_id       
|          | • device.sys_updated_by  
|          | • device.sys_created_by  
|          | • device.primary_email  
|          | • device.schedule     
|          | • device.name         
|          | • device.sys_created_on  
|          | • device.email_address  
|          | • device.active       
|          | • device.phone_number  
|          | • device.sys_mod_count  
|          | • device.sys_updated_on  
|          | • device.user         
|          | • device.sys_meta      
|          | • device.type         |

Make a notification mandatory

To prevent users from turning off or deleting a subscription to a notification, make the notification mandatory.

Role required: admin

1. Navigate to System Notification > Email > Notifications.
2. Open the appropriate notification.
3. Configure the form and add the Mandatory field. This field does not display by default.
4. Select the Mandatory check box.

That notification is now locked in user preferences, preventing the user from removing or unsubscribing to the notification, filtering it, or changing the schedule. In the user’s notification preferences, the control button for the mandatory notification is on and is read only.
Force a notification to be sent

To force a notification to be sent to the specified users, enable forced delivery.

Role required: admin

Forcing a notification means that the relevant users receive the notification, even if they have not subscribed to the notification or have turned off the subscription. In addition, the users receive the notification even if the Notification field is set to Disable.

1. Navigate to System Notification > Email > Notifications.
2. Open the appropriate notification.
3. Configure the form and add the Force delivery field.
4. Select the Force delivery check box.
5. Click Update.

Unlike using the Mandatory option, forcing the delivery does not lock the user's preference or prevent the user from unsubscribing from the notification.

Add an unsubscribe link to an email message

You can add an Unsubscribe link to the bottom of any email body so users can unsubscribe from any subscription-based notification.

Role required: admin

- Paste the following code into an email notification or template:

```java
template.print(' '<a href="' + gs.getProperty('glide.servlet.uri') + 'unsubscribe.do?sysparm_notification=' + email_action.sys_id + '" Unsubscribe</a><br/>
');
```
Create a notification filter

Notification filters enable a user to control the delivery of messages by creating special conditions on multiple tables in a single, reusable filter.

Role required: admin

For example, you can create a filter that controls message delivery when active incidents, problems, and change requests for network issues reach a critical state. Notification filters are available for selection in the Filter field of a user's Notification Preferences form.

**Note:** The system applies the user's filter conditions after the administrator's conditions have been evaluated. If the administrator's conditions fail, the system ignores notification filters.

1. Navigate to System Notification > Email > Notification Filters and create a record.
2. In the Notification conditions related list of the new record, create and submit filter conditions on one or more tables.

3. Repeat the procedure to create additional conditions on other tables for this filter.
Filter device notifications using a schedule

You can associate devices, such as Email, SMS, and Voice, to schedules that define when the devices can and cannot receive notifications.

Role required: admin

Notifications that are triggered outside of the scheduled days and times for the device are not queued up for delivery at a later time. For example, if an administrator selects the Weekdays schedule for an email device, the device receives email notifications triggered between Monday and Friday. If notifications are triggered on Saturday, they are not delivered to the device.

1. Define schedules as needed using System Scheduler > Schedules > Schedules.
2. Add or edit a device.
3. Configure the New Device for System Administrator form and add the Schedule field.
4. In the Schedule field, select the schedule for the device.
5. Click Submit.

Edit the schedule or filter of an existing notification message

Role required: admin

To edit the schedule or filter of an existing notification message:

1. In the Notification Preferences screen, click the message to edit.
   You cannot edit any attributes of a mandatory message.
2. Make the appropriate changes in the form.
3. Click Update.

Notify

Notify allows you to integrate with telephony providers such as the Twilio service to manage phone calls and SMS messages from within your instance.

Notify with the Twilio service

Notify allows you to send and receive phone calls and SMS messages from your instance using the Twilio telephony service.

Notify automatically creates a TwiML application in the Twilio service and configures the application to use the instance as an endpoint. Phone numbers associated with the Twilio subaccount are imported to Notify.

Configure Notify with the Twilio service

You can configure Notify to use the Twilio telephony service.

You must have an SID and authentication token for an active Twilio subaccount.

Important: Ensure that each instance you configure Notify on uses a different Twilio subaccount. Each subaccount specifies a unique SID, authentication token, telephone numbers, and endpoint. Using the same subaccount across multiple instances may cause your Twilio service configuration to be overwritten.
Role required: notify_admin

1. Navigate to Notify > Twilio Configuration.
2. Enter your Account SID.
3. Enter your Auth Token.
4. Click Save.

A read-only list of phone numbers associated with this Twilio subaccount appears. This list displays the phone number, supported capabilities such as voice or SMS, the country for each Twilio number, and the Notify number group the number belongs to.

**Note:** If you buy or release numbers on the Twilio subaccount, open the configuration page again to refresh the list of numbers. Numbers removed from the Twilio service remain as Notify Number records, but with the **Active** field set to false. Use only active phone numbers for inbound or outbound communication.

Phone numbers you import to Notify are stored on the Notify Phone Numbers [notify_number] table. Assign each number to a number group. Number groups allow you to control which workflows run when using the phone numbers in each number group.

### Outbound communication requirements

Outbound communications initiated through Notify, such as phone calls and SMS messages must satisfy certain requirements.

These requirements apply to all outbound communications initiated through Notify, such as by using Notify workflow activities or the Notify JavaScript API.

Invalid numbers prevent Notify workflows from running, and cause an error to be logged. Set the glide.notify.debug property to true to create detailed error logs.

### Recipient number requirements

These requirements apply to any number that receives a Notify phone call or message.

- The number must be E.164 compliant
- The number must be different than the phone number used to initiate the call or message
- The number must not be a phone number used by Notify

### Number groups

Number groups allow you to group Notify phone numbers and share workflows across grouped numbers.

Number groups allow you to provide consistent functionality in different geographic regions using local phone numbers.

Each phone number within Notify has an associated number group. Numbers within a group use the same workflows for handling incoming and outgoing calls and SMS messages.

Numbers associated with a group appear on the **Notify Phone Numbers** related list on the Notify Phone Number Group form.

### Create a number group

Create a number group to organize Notify phone numbers and associate them with inbound and outbound workflows.
Role required: notify_admin

1. Navigate to Notify > Number Groups.
2. Click New.
3. Select workflows for incoming and outgoing calls and SMS messages.
   Incoming call workflows must run on the Notify Call [notify_call] table. Incoming and outgoing SMS
   workflows must run on the Notify Message [notify_message] or Global [global] tables.
4. Click Submit.

After creating a number group, associate numbers with that group to use the selected workflows.

Notify calls

As a Notify administrator, you can view a list of calls made to or from Notify.

To view a list of calls, navigate to Notify > Calls.

Call records are generated automatically when a user calls a Notify number, or when a user makes an
outbound call through Notify. You cannot create call records manually. You can delete call records.

Note: Deleting a call record may result in cascade deletion of related records.

How incoming Notify calls are processed

Notify processes incoming calls using workflow activities.

Any Notify activity that manages incoming phone calls creates a record on the Notify Workflow Activity
[notify_wf_activity] table. Each notify_wf_activity record is associated with a single call. These records
store JSON data detailing the actions to send to the telephony provider.

Notify processes incoming calls in the following way:

1. A person calls a Notify phone number.
2. Notify launches the incoming call workflow associated with that Notify phone number.
3. The workflow reaches a Notify activity and invokes the activity onExecute() function.
4. The activity creates a new notify_wf_activity record detailing any actions to take, with a State value of
   execute.
5. Notify sends the specified actions to the telephony provider.
6. The notify_wf_activity record State changes to processed.
7. The telephony provider sends a response.
8. Response arguments, such as user input or recording info, are stored as JSON data in the
   notify_wf_activity response_args field.
9. The notify_wf_activity State changes to complete.
10. The JSON data from the notify_wf_activity record is copied to the Last action field in the Notify call
    record that triggered the workflow.
11. The workflow invokes the onUpdate() function in executing activities.
12. The activity confirms that the associated notify_wf_activity record has completed, and changes the
    activity state to finished.
13. The workflow transitions to the next activity.
Notify conference calls

Notify allows you to connect multiple callers to a single conference call.

Conference call records are stored on the Notify Conference Call [notify_conference_call] table. Conference call participant records are stored on the Notify Participant [notify_participant] table.

Conference call and participant records are created automatically when starting and connecting to a conference call. Most fields, such as the Duration of a call are set automatically using business rules.

Notify administrators can manage conference call participants, such as by muting or kicking them.

Several conference call workflows are available by default. These workflows can be used for incoming and outgoing calls and incoming SMS messages. Conference call workflows are:

- Notify: (Re)join Conference Call
- Notify: Join Conference Call Via SMS

The number groups Conference Call Group and Notify On Task Group use these conference call workflows by default.

**Mute or kick a conference call participant**

You can mute or kick conference call participants.

Role required: notify_admin

Before starting this procedure, ensure there is an active conference call with one or more participants.

1. Navigate to Notify > Conference Calls.
2. Select a conference call.
3. In the Notify Conference Call Participants related list, select a participant.
4. Click Mute or Kick.

You can unmute a muted participant but cannot add a kicked participant back to the conference call. A kicked participant may rejoin the conference call, such as by calling in again.

Using SMS with Notify

You can send and receive SMS messages using Notify.

You can send SMS messages using the Notify API sendSMS method, or using the send SMS workflow activity.

Inbound and outbound SMS messages are stored on the Notify Messages [notify_message] table.

Notify supports sending concatenated SMS messages. Messages that exceed 160 characters, or 70 unicode characters, are split into multiple messages. This may affect your total telephony cost as each message can incur a cost. Some mobile carries, such as Sprint, do not support concatenated SMS messages.

Notify workflow activities

Notify activities manage calls and SMS messages in Notify.

All Notify workflows that manage a phone call must run on the notify_call table. Call-related Notify activities can be added only to workflows where the Table field value is Notify Call [notify_call].

You can identify if an outbound call was answered by a human or by an answering machine from within a Notify workflow by evaluating the current.is_human variable, such as with an If workflow activity. This variable is set by the telephony provider when an outbound call is answered. This variable is always true for inbound calls.
**Note:** Do not add a **Timer** activity between multiple Notify activities that interact with active phone calls.

**Important:** When creating a Notify workflow, set the workflow **If condition matches** field to **None**. Notify controls which workflow to run based on the configured number groups.

Join Conference Call workflow activity

The **Join Conference Call** connects an incoming or outgoing call to a Notify conference call.

Notify includes the workflows Notify: (Re)join Conference Call and Notify: Join Conference Call Via SMS to demonstrate how to use the **join conference call** activity to connect inbound and outbound calls, and inbound SMS messages to a conference call.

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 755: Input variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced</td>
<td>Select this check box to display advanced configuration options.</td>
</tr>
<tr>
<td>Script</td>
<td>Specify advanced configuration options using JavaScript, such as if the new participant should be muted upon joining the conference call. You can access values from the workflow scratchpad.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. The **join conference call** activity does not specify any conditions by default.

You can add an error condition to this activity. The activity transitions through the error condition if the **conference_call** scratchpad variable is not set.

**Scratchpad entries**

The activity uses the workflow scratchpad to read persistent values.
Table 756: Values read from scratchpad

<table>
<thead>
<tr>
<th>Scratchpad variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow.scratchpad.conference_call</td>
<td>A GlideRecord for a single conference call record. A call processed by this activity is added to this conference call. If this value is not specified, the <strong>join conference call</strong> activity will log an error.</td>
</tr>
<tr>
<td></td>
<td>When initiating an outgoing call workflow using the Notify API <code>call(String notifyPhoneNumber, String toPhoneNumber, GlideRecord conferenceCall)</code> method, this scratchpad value is set automatically to the conference call GlideRecord. For incoming call workflows, or workflows initiated using a different mechanism, you must explicitly set this scratchpad value.</td>
</tr>
</tbody>
</table>

Call workflow activity

The **Call** activity makes outbound phone calls using a Notify workflow.

Input variables

Input variables determine the initial behavior of the activity.

Table 757: Input variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify Number</td>
<td>The Notify phone number to make the call from. When you initiate a call, the outgoing call workflow for the number group associated with this number runs.</td>
</tr>
<tr>
<td>Phone number to call</td>
<td>The E.164-compliant phone number to call.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to determine number to call, and the Notify number to call from instead of using the <strong>Phone number to call</strong> and <strong>Notify Number</strong> variables.</td>
</tr>
<tr>
<td>Script</td>
<td>Define a script that controls which number to call. This script should return a string listing the Notify number <code>sys_id</code>, as well as the phone number to call, such as <code>{notify_number: 'sys_id', phone_number: '+316...'}</code>.</td>
</tr>
</tbody>
</table>

Conditions

The conditions determine which transition comes after this activity. The **call** activity does not specify any conditions by default.
You can add an error condition to this activity. The activity transitions through the error condition if the call could not be set up due to invalid data returned by the advanced script.

Send SMS workflow activity

The **Send SMS** activity sends short text messages to users' phones using Notify.

**Input variables**

Input variables determine the initial behavior of the activity.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Select the Notify phone number to use to send the SMS message.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use JavaScript to determine which numbers to send the message to, and the Notify number to use to send the message.</td>
</tr>
<tr>
<td>To</td>
<td>Select any number of users to send the message to. The user record must have an E.164-compliant phone number or notification device configured for SMS messages.</td>
</tr>
<tr>
<td>To (groups)</td>
<td>Select any number of groups to send the message to. All members of that group with an E.164-compliant phone number or SMS notification device receive the message.</td>
</tr>
<tr>
<td>Message</td>
<td>Enter the message to send. You can add field values from the current record by using the <strong>Select variables</strong> box. You can also add values from the workflow scratchpad. If a field and a scratchpad variable have the same name, the field value is used. Because you can use variables in this message, it is not possible to determine the length of the message at design time. If the activity sends a message that is longer than supported by the telephony provider, the message is truncated and the instance logs a warning.</td>
</tr>
<tr>
<td><strong>Variable</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td><strong>To (script)</strong></td>
<td>Enter a script to determine which numbers to send the message to, and the Notify number to use to send the message. The script should return a JavaScript object with the format <code>{notify_number: \'...sys_id...\', users: [...], groups: [...], numbers: [...]}</code>. Specify the users or groups to send the message to as an array of sys_id values. Specify other numbers as an array of E.164-compliant phone numbers. This field appears only if <strong>Advanced</strong> is selected.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. This activity does not specify any conditions by default.

You can add an error condition to this activity. The activity transitions through the error condition if the Notify number used to make the call is not configured correctly or unable to send SMS messages, or if an error occurs while sending the SMS.

**Forward Call workflow activity**

The **Forward Call** activity forwards a Notify call to an E.164-compliant phone number.

If the person receiving a forwarded call hangs up, the **forward call** activity completes and transitions to the next activity. Any further Notify activities in the workflow run for the caller only.

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 759: Input variables**

<table>
<thead>
<tr>
<th><strong>Variable</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phone number to call</strong></td>
<td>Enter the phone number to forward the call to.</td>
</tr>
<tr>
<td><strong>Timeout (in seconds)</strong></td>
<td>Enter the amount of time to wait for the call to be answered before hanging up.</td>
</tr>
<tr>
<td><strong>Record</strong></td>
<td>Select this check box to record the conversation.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. The **forward call** activity does not specify any conditions by default.
You can add an error condition to this activity. The activity transitions through the error condition if the phone number to call is invalid.

**Input workflow activity**

The **Input** activity creates a phone menu by presenting a list of options on a Notify call.

**Input Variables**

Input variables determine the initial behavior of the activity.

**Table 760: Input Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of digits</td>
<td>Specify the maximum number of digits the caller can enter. A caller can enter fewer digits than the maximum and press the <strong>Finish key</strong> to complete the entry.</td>
</tr>
<tr>
<td>Finish key</td>
<td>Specify the key a caller can press on their phone when finished selecting a menu option.</td>
</tr>
<tr>
<td>Timeout (in seconds)</td>
<td>Specify the amount of time to wait before closing the menu automatically when the caller does not select a menu option.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to build the phone menu, instead of using the activity conditions.</td>
</tr>
<tr>
<td>Script</td>
<td>Define the script to build the phone menu. The script must specify an answer variable as a JavaScript object with the following format:</td>
</tr>
</tbody>
</table>

```javascript
answer = {
  1: {
    "say": "https://some_url.com/options/one.mp3",
    "myCustomData": "some data here"
  },
  2: {
    "say": "type 2 to speak to a representative",
    "language": "en-US",
    "myCustomData": "some more data here"
  }
};
```

The script may specify either a text-to-speech string and language code or the URL of a prerecorded message for each entry. You can also add optional attributes to store related information, such as myCustomData in the example above.
**Conditions**

The conditions determine the transition that comes after this activity.

The input activity does not specify any conditions by default. You must define conditions to build the phone menu. Each condition is one option on the phone menu. Notify reads the text from each condition Name to the caller, up to 100 characters per condition.

You can specify a language for each condition by prefixing the message with the language code, in the format xx-XX:<Message>. For example, add fr-CA: for Canadian French. Available languages are stored on the Notify Language [notify_language] table.

The condition that the activity transitions through depends on the digits entered by the caller. Set the condition **Condition** value to `parseInt(workflow.scratchpad.digits) == <expected digits>`. For example, to transition through a condition when the caller presses the number 3, set the **Condition** to `parseInt(workflow.scratchpad.digits) == 3`.

You can add an error condition to this activity. The activity transitions through the error condition if the advanced script returns an invalid value, or if the text to say for a condition is empty.

**Scratchpad Entries**

The activity uses the workflow scratchpad to write persistent values.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow.scratchpad.digits</td>
<td>The digits entered by the caller, as a string.</td>
</tr>
</tbody>
</table>
| workflow.scratchpad.menu<activity name> | The entire answer variable, if using the advanced script option. You can access this menu from other activities after this activity successfully executes.  
For example, if the activity name is choices, you can access values from the menu using  
```javascript
var previousActivity = "choices";
var choicesMenu = workflow.scratchpad.menu[previousActivity];
var menuItem = choicesMenu[workflow.scratchpad.digits];
// Selects the menu item based on the caller's input.
var selectedValue = menuItem.myCustomData; // get the custom data for the selected menu item.
``` |

**Hangup workflow activity**

The **Hangup** activity disconnects an active Notify phone call.

You can use the **hangup** activity to disconnect only calls that have been answered. Use the **reject** activity to disconnect calls that have not been answered.
Play workflow activity

The **Play** activity plays a sound file on a Notify call.

**Input Variables**

Input variables determine the initial behavior of the activity.

**Table 762: Input Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Enter the URL of a sound file to play. If the URL is inaccessible, or if the audio file mime type is not supported by the telephony provider, the play activity is skipped.</td>
</tr>
<tr>
<td>Loop</td>
<td>Enter the number of times the sound file should play.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. The **play** activity does not specify any conditions by default.

You can create an error condition to handle errors with the sound file. The activity transitions through the error condition if the specified sound file is not available, or has an unsupported mime type. Supported mime types are listed on the Notify Audio MIME Types [notify_mime_type] table.

Record workflow activity

The **Record** workflow activity records audio from a user on a Notify call.

**Input Variables**

Input variables determine the initial behavior of the activity.

**Table 763: Input Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max duration (in seconds)</td>
<td>Enter the maximum duration, in seconds, allowed for a recording.</td>
</tr>
<tr>
<td>Timeout (in seconds)</td>
<td>Enter the amount of time to wait before ending a recording automatically when the caller is silent.</td>
</tr>
<tr>
<td>Finish Key</td>
<td>Specify the key a caller can press on their phone to end the recording.</td>
</tr>
</tbody>
</table>
**Scratchpad Entries**

The activity uses the workflow scratchpad to store persistent values. The **record** activity adds the recording variable to the workflow scratchpad. This variable stores metadata about the recording, such as URL, ID, and duration. You can access the following values from this variable.

**Table 764: Values written to scratchpad**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recording.recordingDuration</td>
<td>The duration of the recording, in seconds.</td>
</tr>
<tr>
<td>recording.from_number</td>
<td>The caller's phone number, including country code.</td>
</tr>
<tr>
<td>recording.notify_number</td>
<td>The Notify phone number used to respond to the call, including country code.</td>
</tr>
<tr>
<td>recording.recordingID</td>
<td>The ID used by the telephony provider to identify the recording.</td>
</tr>
<tr>
<td>recording.recordingURL</td>
<td>The URL from the telephony provider to access the recording.</td>
</tr>
</tbody>
</table>

**Reject workflow activity**

The **Reject** workflow activity rejects an incoming Notify call. You can use the **reject** activity to disconnect only calls that have not yet been answered. Use the **hang up** activity to disconnect calls that have been answered.

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 765: Input variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason</td>
<td>Select a reason for rejecting the call, such as <strong>busy</strong>.</td>
</tr>
</tbody>
</table>

**Say workflow activity**

The **Say** activity plays a message, using text-to-speech, on a Notify call.

**Input variables**

Input variables determine the initial behavior of the activity.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text to say</td>
<td>Specify the text to read.</td>
</tr>
<tr>
<td>Language</td>
<td>Select the language and locale to use when reading text.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to define the language and message,</td>
</tr>
<tr>
<td></td>
<td>instead of using the <code>Text to say</code> and <code>Language</code> values.</td>
</tr>
<tr>
<td>Script</td>
<td>Define a script to set what text is read on the call. The script must</td>
</tr>
<tr>
<td></td>
<td>return a string that defines the language and the text to read. For</td>
</tr>
<tr>
<td></td>
<td>example, to play an English-language message, return `{language: 'en-US',</td>
</tr>
<tr>
<td></td>
<td>text: 'Text to read'}.</td>
</tr>
</tbody>
</table>

**Forward to Notify Client workflow activity**

The **Forward to Notify Client** workflow activity connects a phone call to a Notify WebRTC client.

**Input variables**

Input variables determine the initial behavior of the activity.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Select the user to connect the call to.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to determine which client to connect</td>
</tr>
<tr>
<td></td>
<td>to, instead of using the <code>User</code> variable.</td>
</tr>
<tr>
<td>Script</td>
<td>Define a script that controls which client to connect to. This script</td>
</tr>
<tr>
<td></td>
<td>should return a GlideRecord for a single User [sys_user] record.</td>
</tr>
<tr>
<td>Timeout (in seconds)</td>
<td>Enter the amount of time to wait for the call to be connected before hanging up.</td>
</tr>
<tr>
<td>Record</td>
<td>Select this check box to record the call.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. The **Forward to Notify Client** activity does not specify any conditions by default.

You can add an error condition to this activity. The activity transitions through the error condition if there is an issue with the Notify client.
Queue workflow activity

The **Queue** activity places an active Notify call in a queue.

Add the **Queue** activity to a workflow on the Notify Call [notify_call] table to put the current call on hold. This activity does not specify any input variables.

The queue that the call is added to is given a random ID.

Notify activity event handlers

Notify workflow activities provide JavaScript functions that are invoked automatically when your telephony provider broadcasts certain events.

When a telephony provider such as the Twilio service broadcasts an event related to a Notify call, such as the caller hanging up, the event is processed by the workflow associated with that Notify call. All currently-executing Notify activities in that workflow invoke a JavaScript function associated with that event.

Not all activities that support an event provide a default event handler implementation. You can provide an implementation for those event handlers when creating custom Notify activities.

Creating Notify activities

Default Notify workflow generally perform a single action, such as initiating a phone call. You can create custom Notify workflow activities to implement more complex functionality.

Use event handler functions in each Notify activity definition to design your own activity behavior. For example, you can set the activity result if the caller hangs up by using the **onCompleted** function.

```javascript
onCompleted: function() {
  activity.result = 'call_ended';
},
```

When creating a new Notify workflow activity, follow these design principles.

- Within the **onExecute** event handler function, create a new NotifyAction object. Use this object to create a simple or complex action, then serialize the object to the notify_wf_activity table.
- Within the **onUpdate** event handler function, deserialize the NotifyAction object from the notify_wf_activity table and use this object to perform any other operations.
- Use the other event handler functions and the Notify as needed to implement the desired activity behavior.

Available Notify activity event handlers

When creating Notify workflow activities, you can use event-handler functions to respond to events from a telephony provider.

Not all activities or telephony providers support all events. For example, the **Reject** activity ends the call before the caller connects so the **onNoAnswer** function is never invoked from this activity.

**Note:** When configuring an activity to respond to an event, ensure that your telephony provider supports that event.

The following table describes available event handlers and lists the Notify workflow activities that can invoke each handler.
Table 768: Event handlers

<table>
<thead>
<tr>
<th>Handler</th>
<th>Description</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>onExecute</td>
<td>Runs when the activity first runs.</td>
<td>All call-related activities</td>
</tr>
<tr>
<td>onUpdate</td>
<td>Runs when the call record associated with the workflow is updated.</td>
<td>All call-related activities</td>
</tr>
<tr>
<td>onCompleted</td>
<td>Runs when a call ends.</td>
<td>All call-related activities</td>
</tr>
<tr>
<td>onRinging</td>
<td>Runs when a call starts ringing.</td>
<td>Forward call, call, join conference call, forward to notify client</td>
</tr>
<tr>
<td>onInProgress</td>
<td>Runs when both parties are connected on a call.</td>
<td>Forward call, call, join conference call, forward to notify client</td>
</tr>
<tr>
<td>onNoAnswer</td>
<td>Runs when a call is not answered.</td>
<td>Forward call, call, join conference call, forward to notify client</td>
</tr>
<tr>
<td>onBusy</td>
<td>Runs when a called number is busy.</td>
<td>Forward call, call, join conference call, forward to notify client</td>
</tr>
<tr>
<td>onFailed</td>
<td>Runs when an error occurs.</td>
<td>All call-related activities</td>
</tr>
</tbody>
</table>

Notify languages

Notify supports multiple languages when using text-to-speech.

For example, when using the input or say workflow activities, you can specify which language to use when reading the text.

The list of available languages is stored on the Notify Language [notify_language] table. By default, languages supported by all Notify telephony providers are available. You can add additional languages if they are supported by your specific telephony provider.

Notify on task

Notify allows you to send SMS alerts and initiate conference calls from any task record.

Notify on task functionality is available for all tables that extend the Task table, such as Incident, Problem, or Change.

All calls and messages made from tasks are initiated using a single Notify phone number. You can configure which phone number is used by setting the property glide.notify.task.phone_number.

Send an SMS message from a task

You can use Notify to send SMS alerts from a task record.

Role required: itil
Notify must be set up before you can use Notify on task. Ensure there are Notify phone numbers in number groups with SMS workflows. Also ensure the property glide.notify.task.phone_number is set to one of these Notify phone numbers.

1. Navigate to any task record, such as an incident or change.
2. Click the Send SMS related link.
3. Select one or more Recipients to send the SMS to.
4. Enter the Message to send.
   The task record number is added to the message automatically.
5. Click Send.
   Any SMS records created from a task are associated with that task. You can view associated SMS messages from the task record in the SMS Messages related list. You may need to configure the form to add the related list.

Start a conference call from a task

You can use Notify to start conference calls from a task record.

Role required: itil

Notify must be set up before you can use Notify on task. Ensure there are Notify phone numbers in number groups with phone call workflows. Also ensure the property glide.notify.task.phone_number is set to one of these Notify phone numbers.

1. Navigate to any task record, such as an incident or change.
2. Click the Start conference call related link.
3. Select one or more Recipients to send the SMS to.
4. Click Start.
   Any conference call records created from a task are associated with that task. You can view associated conference calls from the task record in the Conference calls related list. You may need to configure the form to add the related list.

Notify with incident alert

You can use Notify with Incident Alert to send SMS messages or start conference calls based on incident alerts.

Notify with incident alert requirements

Certain configuration steps are required to use Notify with Incident Alert.

To use notify with incident alert:

- Both Notify and Incident Alert must be active.
- Notify must be configured and able to initiate and receive phone calls and SMS messages.
- The property com.snc.iam.notify_number must be set to an active Notify number.
- This Notify number must belong to a number group configured with inbound and outbound conference call workflows.

Note: The Conference Call Group number group is configured with conference call workflows by default.

Launch a conference call

As part of processing an incident alert, a conference call can be created between involved users.
Call participants can include:
- Those users who have been assigned specific responsibilities.
- Any required ad-hoc user contacts.
- Other involved parties who are not recorded as users, such as third-party contacts.

**Note:** Only one conference call at a time can be active for each incident.

1. Navigate to Incidents > Open.
2. Open the relevant incident alert.
3. Click the Initiate Conference Call related link.
4. Within the dialog box that appears, select the participants for the conference.

The dialog box displays the recommended and selected participants for the conference. All users from the User Contacts list in the incident alert are selected by default. If a rotation schedule exists for the Group Contacts, the primary and secondary on-call resources are shown in the Recommended list. This way, the current on-call persons can quickly be invited to join the conference call. Calls are placed to the number in the Mobile phone field on the user record. If that information is blank, the user cannot be contacted through Notify. The mobile phone number has to be an E.164 compliant phone number. If the phone number is a local number, without the + prefix, the number will be retrieved based on the user's location and, if possible, converted into a valid E.164 number.

5. To select ad-hoc participants, do one of the following:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click the reference lookup icon, and select the relevant user.</td>
<td>Click Add to selected.</td>
</tr>
<tr>
<td>Enter the participant’s phone number in the field beside the telephone icon.</td>
<td>Click Add to selected.</td>
</tr>
</tbody>
</table>

6. After the participant list is finalized, click **OK**.
   
The conference call starts and a **Conference call initiated** message is displayed at the top of the Incident Alert form. Each user is called and can accept the call to join the conference. Several response types are possible from users invited to join the conference call, apart from **Accepted**.

7. Click the **Conference call initiated** message to see details of that conference call.
   
   When the final participant leaves the conference, the conference call closes.

**Note:** VoIP phone systems, which do not use touch tone phones, may encounter issues with recognizing key presses. To avoid problems, ensure that conference call users use touch tone phones, or configure your VoIP system settings to recognize key presses, as described in your VoIP system documentation.

**Viewing conference call information**

Conference calls are listed as system activities in the **Activity** section of the Incident Alert form and also are listed in the **Conference Calls** related list.
Figure 793: Conference bridge history
Send an SMS notification

Notify sends an SMS to the users defined as default contacts for the alert.

When you create a new incident alert, Notify sends an SMS notification to the users defined as default contact responsibilities for the alert.

This text message is sent to the user's mobile phone number:

IA<number>: a <Severity> severity <Event Type> incident alert for <CI Name> has been opened.

Administrators can modify the content of this message by editing the SMS on new Incident Alert business rule.

Notify with on-call scheduling

When using both Notify and On-call scheduling you can send scheduling notifications as SMS messages, in addition to emails.

Notify with on-call scheduling requirements

Certain configuration steps are required to use Notify with On-call scheduling.

- You must add at least one Notify phone number to the On-Call Group number group. This group is configured by default to handle inbound SMS responses that accept or reject an on-call assignment.
- You must configure any outbound SMS workflows, such as the On-Call: Assign by Acknowledgement workflow to use this Notify phone number to send SMS messages. Edit all Send SMS activities in these workflows to use this number as the From value.

Set up Notify with on-call scheduling

You can configure on-call scheduling to use Notify functionality to send SMS alerts and assign users to tasks based on the SMS responses.

You must have activated the Notify and On-Call Scheduling plugins on your instance.

Role required: admin

Set up Notify with On-call scheduling for the first time, or migrate from On-call scheduling with NotifyNow.

1. Navigate to Notify > Numbers.
2. Select the phone number you want to use to send and receive on-call scheduling messages.
3. In the Notify group field, select the On-Call Group.
   The On-Call Group is configured with the on-call workflow by default.
4. Navigate to Workflow > Workflow Editor.
5. Select the On-Call: Assign by Acknowledgement workflow.
   If migrating from NotifyNow, and you have previously customized this workflow, you must import the new version of the workflow manually. See KB0551603 for more information.
6. Click the workflow actions icon and select Checkout.
7. Select a Send SMS activity.
8. In the To (script) field, modify the getRecipientsAndNumberToSendFrom() function and uncomment the line //notify_number : getNotifyNumber(),.
9. Perform one of the following actions.
### Option | Description
--- | ---
**Specify a static number** | Within the `getRecipientsAndNumberToSendFromChange()` function, change `getNotifyNumber()` to the `sys_id` of the Notify phone number you selected in step 2.

**Specify a number procedurally** | Update the `getNotifyNumber()` function to return the `sys_id` of the Notify phone number you selected in step 2.

10. Repeat the previous 2 steps for each **Send SMS** activity on the workflow.

11. Click the workflow actions icon and select **Publish**.

12. Close the workflow editor interface.

13. Navigate to **On-Call Scheduling > Trigger Rules**.

14. Select a trigger rule that launches an assignment workflow.

15. In the **Trigger action** field, select **Workflow**.

16. In the **Trigger workflow** field, select the On-call: Assign by Acknowledgement workflow.

17. Click **Update**.

18. Repeat the previous 3 steps for each assignment trigger rule.
   - Each time the conditions for one of these trigger rules is met, the On-call: Assign by Acknowledgement workflow runs.

---

**On-call: assign by acknowledgement workflow**

The On-Call: Assign by Acknowledgement workflow is provided with Notify.

The workflow uses data from the escalation settings of rotas and rosters. Depending on these settings, the workflow iterates through the defined escalation chain and sends notifications by SMS or email to users asking them for incident assignment.

The workflow respects time-off as specified in the rosters. People who have time-off are not included in the escalation chain and no notifications are sent to them.

When you install both On-call scheduling and Notify, the `message_number` column is added to the Notify Messages [notify_messages] table to track responses to on-call assignment requests. This column indicates if the contacted user accepted or rejected the assignment.

Before you can send notifications, you must define trigger rules. Trigger rules determine the conditions that must be met before a notification is sent and what action must be taken.

---

**Controlling the on-call communication channel with Notify**

You can configure On-call scheduling with Notify to always use a user's preferred communication channel.

If **Force communication channel** is specified in the **Escalation settings** for rosters, the preferred user device is used, either SMS or email.

If the preferred device is SMS, and the on-call member does not have an SMS device defined, the user is not contacted even if the user has an email address. When forcing a communication channel on an escalation level does not succeed, no further communication attempts are made. The fact that the user could not be reached is logged.

The setting **Force communication channel** is only available if Notify is installed.
Key differences between on-call scheduling with NotifyNow and with Notify

There are several key differences between on-call scheduling with NotifyNow and on-call scheduling with Notify.

Tracking responses to questions

The On call: Assign by Acknowledgement workflow no longer uses the Notify Question tables to track assignment responses. When you install On-call scheduling with Notify, the message_number column is added to the Notify Messages [notify_messages] table to track responses to on-call assignment requests. This column indicates if the contacted user accepted or rejected the assignment.

Scripting Notify

You can use scripts with Notify to interact with calls and SMS messages, or to provide a custom client interface.

Notify

The Notify API allows you to interact with Notify calls and SMS messages using scripts.

Notify - getPhoneNumbers()
Get all phone numbers available to Notify, as an array.

Table 769: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 770: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>An array of NotifyPhoneNumber objects, each object representing one phone number available to Notify.</td>
</tr>
</tbody>
</table>

```javascript
// instantiate notify
var notify = new SNC.Notify();

// get all available phone numbers
var phoneNumbers = notify.getPhoneNumbers();

// iterate over phone numbers
for (var i = 0; i < phoneNumbers.size(); i++) {
  var number = phoneNumbers.get(i);
  //perform any actions using each phone number
}
Notify - call(String notifyPhoneNumber, String toPhoneNumber)
Make a call to an E.164-compliant phone number.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifyPhoneNumber</td>
<td>String</td>
<td>The phone number to make the call from. This number appears as the caller ID.</td>
</tr>
<tr>
<td>toPhoneNumber</td>
<td>String</td>
<td>The phone number to call.</td>
</tr>
</tbody>
</table>

Table 772: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var from = '+14048007337';
var to = '+31646810495';

// set up call
new SNC.Notify().call(from, to);
```

Notify - call(String notifyPhoneNumber, String toPhoneNumber, GlideRecord conferenceCall)
Call a number to add that number to an active conference call.

This method is intended specifically for conference calls. To initiate calls between only two participants, use the call(String notifyPhoneNumber, String toPhoneNumber) method instead.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifyPhoneNumber</td>
<td>String</td>
<td>The Notify phone number to make the call from. When you initiate a call, the outgoing call workflow for the number group associated with this number runs. Ensure this workflow includes a join conference call activity to connect the user to the conference call.</td>
</tr>
<tr>
<td>toPhoneNumber</td>
<td>String</td>
<td>The phone number to call. Called numbers are added to the conference call.</td>
</tr>
</tbody>
</table>
### Table 774: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var notify = new SNC.Notify();
var from = '+14041234567';
var participants = ['#31612345678', '+31623456789', '+31687654321'];

// set up a conference call
var conferenceCall = notify.conferenceCall();

// set up the outbound calls for all conference call participants
for (var i in participants) {
    var to = participants[i];
    notify.call(from, to, conferenceCall);
}

// feedback
gs.log(gs.getMessage('set up a conference call with number {0} and (re)join code: {1}',
                        [ conferenceCall.getValue('number'),
                          conferenceCall.getValue('code') ]));
```

**Notify - sendSMS(NotifyPhoneNumber notifyPhoneNumber, String toPhoneNumber, String messageBody)**

Send an SMS text message to an E.164-compliant phone number.

This function creates a new record on the Notify Message [notify_message] table.

### Table 775: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifyPhoneNumber</td>
<td>NotifyPhoneNumber</td>
<td>The Notify phone number to send this SMS from.</td>
</tr>
<tr>
<td>toPhoneNumber</td>
<td>String</td>
<td>An E.164-compliant phone number to send the SMS to.</td>
</tr>
<tr>
<td>messageBody</td>
<td>String</td>
<td>The SMS text.</td>
</tr>
</tbody>
</table>
Table 776: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The sys_id of the Notify Message [notify_message] record created by this function.</td>
</tr>
</tbody>
</table>

Notify - sendSMS(NotifyPhoneNumber notifyPhoneNumber, String toPhoneNumber, String messageBody, GlideRecord source)
Send an SMS text message to an E.164-compliant phone number.
This function creates a new record on the Notify Message [notify_message] table and associates it with the source record.

Table 777: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifyPhoneNumber</td>
<td>NotifyPhoneNumber</td>
<td>The Notify phone number to send this SMS from.</td>
</tr>
<tr>
<td>toPhoneNumber</td>
<td>String</td>
<td>An E.164-compliant phone number to send the SMS to.</td>
</tr>
<tr>
<td>messageBody</td>
<td>String</td>
<td>The SMS text.</td>
</tr>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The source record that prompted this SMS message, such as an incident.</td>
</tr>
</tbody>
</table>

Table 778: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The sys_id of the Notify Message [notify_message] record created by this function.</td>
</tr>
</tbody>
</table>

Notify - getChildCallIDs(GlideRecord callRecord)
Get the sys_id values of calls that are children of a specified call.
Any call started by forwarding another call, such as with the Forward workflow activity, is considered a child of the original call. The original call is the parent call.
Table 779: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callRecord</td>
<td>GlideRecord</td>
<td>A record on the Notify Call [notify_call] table.</td>
</tr>
</tbody>
</table>

Table 780: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>The sys_id values of all calls that are children of the specified call.</td>
</tr>
</tbody>
</table>

```javascript
var callRecord = new GlideRecord('notify_call');
callRecord.get("0f4f5863ff13310014ecffffffffff28");
var notify = new SNC.Notify();
var childCallIDs = notify.getChildCallIDs(callRecord);
for(var callID in childCallIDs)
  //perform any operations with the child callID values
```

Notify - `getParentCallID(GlideRecord callRecord)`
Get the sys_id of a specified call's parent call.

Any call started by forwarding another call, such as with the Forward workflow activity, is considered a child of the original call. The original call is the parent call.

Table 781: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callRecord</td>
<td>GlideRecord</td>
<td>A record on the Notify Call [notify_call] table.</td>
</tr>
</tbody>
</table>

Table 782: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The sys_id of the parent call record.</td>
</tr>
</tbody>
</table>

```javascript
var callRecord = new GlideRecord('notify_call');
callRecord.get("0f4f5863ff13310014ecffffffffff28");
var notify = new SNC.Notify();
var parentCallID = notify.getParentCallID(callRecord);
```
Notify - `getTokens()`
Get client tokens for any installed telephony drivers for use in WebRTC or mobile clients.
This function uses the currently logged-in user record as the client.

### Table 783: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 784: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The client tokens, as a JSON string.</td>
</tr>
</tbody>
</table>

```javascript
// get Notify Client Tokens per active Notify Driver for the currently logged in user
var json = new SNC.Notify().getTokens();

// parse the json that was return into a tokens object
var tokens = JSON.parse(json);

// log line
gs.log('Notify Client Tokens for the currently logged in user');

// iterate over the driver tokens
for (var driver in tokens) {
    gs.log(driver + ' Driver token: ' + tokens[driver]);
}
```

Output:

Notify - `getTokens(GlideRecord record)`
Get client tokens for any installed telephony drivers for use in WebRTC or mobile clients.

### Table 785: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>GlideRecord</td>
<td>A record used to generate the client tokens.</td>
</tr>
</tbody>
</table>

### Table 786: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The client tokens. The value of the string depends on the GlideRecord parameter.</td>
</tr>
</tbody>
</table>
This example demonstrates getting Notify client tokens for every Notify group.

```javascript
// instantiate Notify
var notify = new SNC.Notify();

// get all Notify Groups
var notifyGroup = new GlideRecord("notify_group");
notifyGroup.query();

// iterate over all notify groups
while (notifyGroup.next()) {
  // generate Notify Client tokens per active Notify Driver
  for this group
    var json = notify.getTokens(notifyGroup);
    var tokens = JSON.parse(json);

    for (var driver in tokens) {
      gs.log(gs.getMessage("Notify Client token for
{0} driver and Notify Group '{1}': {2}",
[driver, notifyGroup.getValue('name'), tokens[driver]]));
    }
}
```

**Notify - getAvailableClients(String notifyNumber)**

Get a list of client sessions that are available to receive calls.

**Table 787: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifyNumber</td>
<td>String</td>
<td>Any valid Notify phone number.</td>
</tr>
</tbody>
</table>

**Table 788: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>An array of JSON objects with the following format:</td>
</tr>
<tr>
<td></td>
<td>[{ sys_id: &quot;...&quot;, // user's</td>
</tr>
<tr>
<td></td>
<td>sys_id</td>
</tr>
<tr>
<td></td>
<td>name: &quot;...&quot; // user's name }</td>
</tr>
</tbody>
</table>

**Notify - conferenceCall()**

Create a new conference call GlideRecord.
Table 789: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 790: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideRecord</td>
<td>A new Notify Call [notify_call] record for a conference call.</td>
</tr>
</tbody>
</table>

```javascript
var notify = new SNC.Notify();
var from = '+14041234567';
var participants = ['+31612345678', '+31623456789', '+31687654321'];

// set up a conference call
var conferenceCall = notify.conferenceCall();

// set up the outbound calls for all conference call participants
for (var i in participants) {
    var to = participants[i];
    notify.call(from, to, conferenceCall);
}

// feedback
gs.log(gs.getMessage('set up a conference call with number {0} and (re)join code: {1}',
    [ conferenceCall.getValue('number'),
    conferenceCall.getValue('code') ]));
```

**Notify - queueCall(GlideRecord callRecord)**

Put a call into a queue.

Resume a queued call using the dequeueCall method.

Table 791: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callRecord</td>
<td>GlideRecord</td>
<td>A GlideRecord object on the Notify Call [notify_call] table with the call you want to put on hold.</td>
</tr>
</tbody>
</table>
Table 792: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var call = new GlideRecord('notify_call');
call.get('<call record sys_id>');
if (call.isValid()) {
    new SNC.Notify().queueCall(call);
}
```

**Notify - dequeueCall(GlideRecord callRecord)**
Resume a call after it was put in a queue.
Use this method to resume calls that were put in a queue with the queueCall method.

Table 793: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callRecord</td>
<td>GlideRecord</td>
<td>A GlideRecord object on the Notify Call [notify_call] table with the held call you want to resume.</td>
</tr>
</tbody>
</table>

Table 794: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var call = new GlideRecord('notify_call');
call.get('some sys_id');
if (call.isValid()) {
    new SNC.Notify().dequeueCall(call);
}
```

**Notify - kick(GlideRecord participant)**
Kick a specified user from a Notify conference call.

Table 795: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>GlideRecord</td>
<td>A GlideRecord object containing the Notify Participant [notify_participant] to kick from the conference call.</td>
</tr>
</tbody>
</table>
Table 796: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var participant = new GlideRecord('notify_participant');
participant.get('<sys_id>');
if (participant.isValid()) {
    new SNC.Notify().kick(participant);
}
```

*Notify - forwardCall(GlideRecord call, String destination, String dtmf)*
Forward a call to connect that call with a different recipient.

Table 797: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>call</td>
<td>GlideRecord or String</td>
<td>A Notify call record, or the telephony provider call ID.</td>
</tr>
<tr>
<td>destination</td>
<td>GlideRecord or String</td>
<td>A Notify phone number record, or an E.164-compliant phone number.</td>
</tr>
<tr>
<td>dtmf</td>
<td>String</td>
<td>A DTMF code to play upon connection.</td>
</tr>
</tbody>
</table>

Table 798: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

**NotifyAction**

The NotifyAction API allows you to define actions to send to a telephony provider.
You add actions to a NotifyAction object by calling the respective `add` function for each type of action.
Each `add` function returns an Action object, such as a SayAction object for the `addSay()` function. Refer to each method example for information about returned objects.

*NotifyAction - addDial()*
Make an outbound call.
Table 799: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 800: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DialAction</td>
<td>The action added to the NotifyAction object. Use the DialAction object to define.</td>
</tr>
</tbody>
</table>

```javascript
var action = new SNC.NotifyAction();
var dial = action.addDial();
dial.setRecord(activity.vars.record);
dial.setClientRecord(activity.vars.user, "sys_user");
```

**NotifyAction - addGather()**

Present an interactive phone menu to the user.

Table 801: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 802: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GatherAction</td>
<td>The action added to the NotifyAction object. Use the GatherAction object to define the menu settings and options to present to the user.</td>
</tr>
</tbody>
</table>

```javascript
// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// present the user with a menu
var gather = notifyAction.addGather();
gather.setNumberOfDigits(1);    // the user can type 1 digit
gather.setFinishKey('#');      // # or *, useful for > 1 digits
gather.setTimeout(10);          // time to enter answer, in seconds

// add first menu item
var usSay = gather.addSay();
usSay.setText('Press 1 for english');
usSay.setLanguage('en-US');
```
// add second menu item
var nlSay = gather.addSay();
nlSay.setText('Kies 2 voor Nederlands');
nlSay.setLanguage('nl-NL');

// add third menu item
var frSay = gather.addSay();
frSay.setText('Choisissez 3 pour le français.);
frSay.setLanguage('fr-FR');

// and finish off with an applause
var play = gather.addPlay();
play.setURL('http://www.wavsource.com/
snds_2015-04-12_5971820382841326/sfx/applause_y.wav');

NotifyAction - addHangUp()
End an active phone call.

Table 803: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 804: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HangUpAction</td>
<td>The action added to the NotifyAction object.</td>
</tr>
</tbody>
</table>

// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// hang up
notifyAction.addHangUp();

NotifyAction - addQueue()
Queue the call, putting it on hold.

Table 805: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 806: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueAction</td>
<td>The action added to the NotifyAction object. Use the QueueAction object to define the queue name, and queueing or dequeueing behavior.</td>
</tr>
</tbody>
</table>

```javascript
// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// queue the call
var queue = notifyAction.addQueue();
queue.setName('my queue');
```

```javascript
// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// dequeue the call
var queue = notifyAction.addQueue();
queue.setDequeue(true);
```

**NotifyAction - addPlay()**

Play an audio file on the call.

Table 807: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 808: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlayAction</td>
<td>The action added to the NotifyAction object. Use the PlayAction object to define the audio file URL and number of times to loop the audio.</td>
</tr>
</tbody>
</table>

```javascript
// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// add a play action
var play = notifyAction.addPlay();
play.setURL('http://www.moviesounds.com/2001/imsorry.wav');
play.setLoop(1);
```
**NotifyAction - addReject()**
Reject an incoming call.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 810: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RejectAction</td>
<td>The action added to the NotifyAction object. Use the RejectAction object to define the reason for rejecting the call.</td>
</tr>
</tbody>
</table>

```javascript
// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// reject the call
var rejectAction = notifyAction.addReject();
rejectAction.setReason('busy'); // 'busy' or 'rejected'
```

**NotifyAction - addSay()**
Use text-to-speech to read text on the call.

Multiple languages are supported with text-to-speech. Available languages depend on the telephony provider.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 812: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SayAction</td>
<td>The action added to the NotifyAction object. Use the SayAction object to define the text and language to read.</td>
</tr>
</tbody>
</table>

This example demonstrates reading text in several languages.

```javascript
// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// add a say action to say something in US English
```
var usSay = notifyAction.addSay();
usSay.setText('Welcome. I can speak english');
usSay.setLanguage('en-US');

// add a say action to say something in Dutch
var nlSay = notifyAction.addSay();
nlSay.setText('Ik spreek ook vloeiend nederlands');
nlSay.setLanguage('nl-NL');

// and german
var deSay = notifyAction.addSay();
deSay.setText('Und ich kann auch deutsch sprechen');
deSay.setLanguage('de-DE');

NotifyAction - addSMS()
Send an SMS message.

When using this function with an active call, you do not need to call the setTo function on the returned SMSAction object. The SMS is automatically sent to the caller.

Table 813: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 814: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMSAction</td>
<td>The action added to the NotifyAction object. Use the SMSAction object to define the message text and the phone number to send the message to.</td>
</tr>
</tbody>
</table>

// instantiate NotifyAction
var notifyAction = new SNC.NotifyAction();

// define where to send the sms to
var number = new GlideElementPhoneNumber();
number.setPhoneNumber('+31612345678', true);

// add a SMS action
var sms = notifyAction.addSMS();
sms.setMessage('Lorem ipsum dolor sit amet, consectetur adipiscing elit. ');
sms.setTo(number);

NotifyAction - fromJson(String json)
Deserializa a NotifyAction object from a JSON string.
### Table 815: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>json</td>
<td>String</td>
<td>A JSON string representation of a NotifyAction object.</td>
</tr>
</tbody>
</table>

### Table 816: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

This example demonstrates deserializing a NotifyAction object.

```javascript
var json = ".... some json obtained from toJson ....";

// instantiate notify action
var notifyAction = new SNC.NotifyAction();

// deserialize and reconstruct the notify action instance
notifyAction.fromJson(json);
```

This example demonstrates both serializing and deserializing a NotifyAction object.

```javascript
// instantiate notify action
var notifyAction = new SNC.NotifyAction();

// add a queue
var queue = notifyAction.addQueue();
queue.setName('myQueueName');
queue.setDequeue(false);

// serialize to json
var json = notifyAction.toJson();
gs.log('serialization result: ' + json);

// instantiate a new notify action
var newAction = new SNC.NotifyAction();

// deserialize the json generated above
newAction.fromJson(json);

// serialize the new object and log the result
newJson = newAction.toJson();
gs.log('new serialization result: ' + newJson);
gs.log('the same: ' + (json == newJson));
```

Output: *** Script: serialization result: {"fClassName":"NotifyAction","fActions": [{"fClassName":"QueueAction","fDequeue":true,"fQueueName":"myQueueName"}]} ***

*** Script: new serialization result: {"fClassName":"NotifyAction","fActions": [{"fClassName":"QueueAction","fDequeue":true,"fQueueName":"myQueueName"}]} ***

*** Script: the same: true ***
**NotifyAction - toJson()**

Serialize the NotifyAction object to a JSON string.

**Table 817: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 818: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>A JSON representation of this NotifyAction object.</td>
</tr>
</tbody>
</table>

This example demonstrates serializing a NotifyAction object.

```javascript
// instantiate notify action
var notifyAction = new SNC.NotifyAction();

// add one or more notify actions
// ...

// and serialize to json
var json = notifyAction.toJson();
```

This example demonstrates both serializing and deserializing a NotifyAction object.

```javascript
// instantiate notify action
var notifyAction = new SNC.NotifyAction();

// add a queue
var queue = notifyAction.addQueue();
queue.setName('myQueueName');
queue.setDequeue(false);

// serialize to json
var json = notifyAction.toJson();
gs.log('serialization result: ' + json);

// instantiate a new notify action
var newAction = new SNC.NotifyAction();

// deserialize the json generated above
newAction.fromJson(json);

// serialize the new object and log the result
newJson = newAction.toJson();
gs.log('new serialization result: ' + newJson);
gs.log('the same: ' + (json == newJson));
```

Output: *** Script: serialization result: {"fClassName":"NotifyAction","fActions": [{"fClassName":"QueueAction","fDequeue":true,"fQueueName":"myQueueName"}]}***

*** Script: new serialization result: {"fClassName":"NotifyAction","fActions":"}***
NotifyAction - addConference()
Add a call to a Notify conference call.

Table 819: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 820: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConferenceAction</td>
<td>The action added to the NotifyAction object. Use the ConferenceAction object to define the conference call name, and the behavior of the conference call when a participant joins or leaves.</td>
</tr>
</tbody>
</table>

NotifyPhoneNumber

The NotifyPhoneNumber API allows you to query information about a Notify phone number.

NotifyPhoneNumber - getDialCode()
Get the international dialing code for a Notify phone number.

Table 821: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 822: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>International phone code for a country.</td>
</tr>
</tbody>
</table>
Table 823: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 824: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The ID of the number within the telephony provider.</td>
</tr>
</tbody>
</table>

NotifyPhoneNumber - `getNumber()`
Get the numerical phone number for a NotifyPhoneNumber.

Additional information about the method that does not belong in the short description.

Table 825: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 826: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>An E.164-compliant phone number.</td>
</tr>
</tbody>
</table>

NotifyPhoneNumber - `getOwner()`
Get the telephony provider associated with this phone number.

Table 827: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 828: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>The telephony provider associated with the number, such as the Twilio service.</td>
</tr>
</tbody>
</table>

NotifyPhoneNumber - getTerritory()
Get the country associated with the phone number.

Table 829: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 830: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The name of the country the phone number belongs to.</td>
</tr>
</tbody>
</table>

NotifyPhoneNumber - supportsConferenceCall()
Determine if the Notify phone number supports conference calling.
Additional information about the method that does not belong in the short description.

Table 831: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 832: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if the Notify phone number supports conference calling.</td>
</tr>
</tbody>
</table>
Output:

NotifyPhoneNumber - supportsIncomingPhoneCall()
Determine if the Notify phone number supports receiving phone calls.

Table 833: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 834: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if the Notify phone number supports receiving incoming phone calls.</td>
</tr>
</tbody>
</table>

NotifyPhoneNumber - supportsIncomingSMS()
Determine if the Notify phone number supports receiving SMS messages.

Table 835: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 836: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if the Notify phone number supports receiving incoming MMS messages.</td>
</tr>
</tbody>
</table>

NotifyPhoneNumber - supportsOutgoingPhoneCall()
Determine if the Notify phone number supports initiating phone calls.

Table 837: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 838: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if the Notify phone number supports initiating outgoing phone calls.</td>
</tr>
</tbody>
</table>

*NotifyPhoneNumber - supportsOutgoingSMS()*
Determine if the Notify phone number supports sending SMS messages.

Table 839: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 840: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if the Notify phone number supports sending SMS messages.</td>
</tr>
</tbody>
</table>

*NotifyPhoneNumber - supportsRecording()*
Determine if the Notify phone number supports recording phone calls.

Table 841: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 842: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if the Notify phone number supports recording phone calls.</td>
</tr>
</tbody>
</table>
**NotifyPhoneNumber - supportsWebRTC()**
Determine if the Notify phone number supports calls to a browser, such as in a WebRTC implementation.

<table>
<thead>
<tr>
<th>Table 843: Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 844: Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>boolean</td>
</tr>
</tbody>
</table>

**Notify Client**

The Notify Client API allows you to use Notify telephony functionality, such as making and receiving calls, from a web browser.

Several Notify Client methods take a callback function as a parameter. Because Notify Client calls are made asynchronously, these methods cannot return a value directly. Use the callback function to parse the returned data, such as by assigning variables or making other API calls.

**Notify Client - Client(Object notifyConfig)**

Instantiate a new Notify WebRTC Client object.

<table>
<thead>
<tr>
<th>Table 845: Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>notifyConfig</td>
</tr>
</tbody>
</table>

```javascript
var notifyConfig = {
    vendor: SNC.Notify.Vendor.TWILIO,
    callerId: 'xyz',
    autoReconnect: true,
    onReady: function() {},
    onOffline: function() {},
    onError: function( message ) {},
    onConnect: function( status ) {},
    onDisconnect: function() {},
    onIncoming: function( {from, to, callSid} ) {},
    onOutgoing: function( callSid ) {},
    onAccept: function() {},
    onMute: function() {},
    onUnmute: function() {},
    onCancel: function() {}
};
```
Notify Client - `call(Object identifier)`

Call a specified phone number or the phone number associated with a specified user.

### Table 846: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Object</td>
<td>Enter a JSON object that provides either the phone number to call or the sys_id of a user record to get the phone number from.</td>
</tr>
</tbody>
</table>

**Note:** If you provide both a phone number and user sys_id, only the phone number is used.

### Table 847: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

This example demonstrates passing a phone number as the function parameter.

```javascript
notifyClient.call({
    phoneNumber: "+18001112223"
});
```

This example demonstrates passing a user record sys_id as the function parameter.

```javascript
notifyClient.call({
    userId: "6816f79cc0a8016401c5a33be04be441"
});
```

Notify Client - `hangupCall()`

End the current call.
This example demonstrates mapping a client function to an interface button using jQuery.

```javascript
$j("#pickupCallBtn").on("click", function() {
    notifyClient.hangupCall();
});
```

**Notify Client - pickupCall()**
Brief description of the method.

Additional information about the method that does not belong in the short description.

**Table 850: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 851: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

**Notify Client - mute(Boolean muted)**
Mute or unmute the current client.

Additional information about the method that does not belong in the short description.

**Table 852: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>muted</td>
<td>Boolean</td>
<td>Set this value to true to mute the current client, false to unmute.</td>
</tr>
</tbody>
</table>
**Notify Client - forwardCall(Object argument)**
Forward the current call to a different phone number or Notify client session.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>argument</td>
<td>Object</td>
<td>A JavaScript object detailing the number or Notify client to forward the call to.</td>
</tr>
</tbody>
</table>

This example demonstrates forwarding a call to a different phone number. The dtmf attribute allows you to send DTMF dial tones to the receiving number.

```javascript
var arg = {
  type: "number",
  id: "+17012345678",
  dtmf: "1234"
};
client.forwardCall(arg);
```

This example demonstrates forwarding a call to a different Notify client.

```javascript
var arg = {
  type: "userId",
  id: "6816f79cc0a8016401c5a33be04be441"
};
client.forwardCall(arg);
```

**Notify Client - sendDtmf(String digits)**
Send one or more DTMF-valid digits over the current call.
Table 856: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>digits</td>
<td>String</td>
<td>One or more DTMF-valid digits.</td>
</tr>
</tbody>
</table>

Table 857: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

Notify client event handlers

Notify Client event handlers allow you to define scripted behavior for responding to events from a telephony provider such as the Twilio service.

Event handlers

A communication between the Notify WebRTC Client and a telephony provider is asynchronous. The client function calls, such as `notifyClient.call()` do not wait for a response from the telephony provider. Instead, event handler methods define how to respond to certain events from the telephony provider.

You can create event handler implementations when creating the Notify Client configuration object. Fully define the handlers in the configuration object before instantiating a Notify Client object.

onReady

The `onReady` function runs after you call the `init` function on the client object. The `onReady` function indicates that the WebRTC session is ready.

onOffline

The `onOffline` function runs if the WebRTC session is not active.

onError

The `onError` function runs if the client encounters an error.

This function exposes one parameter. Use this parameter when implementing the event handler function.

Table 858: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>String</td>
<td>The error message text.</td>
</tr>
</tbody>
</table>
**onConnect**

The `onConnect` function runs when the client receives a call connection event for an incoming or outgoing call.

This function exposes one parameter. Use this parameter when implementing the event handler function.

**Table 859: Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String</td>
<td>The call status, as a string constant.</td>
</tr>
</tbody>
</table>

*Note:* When evaluating call status, use the constants provided by `SNC.Notify.Status`.

```javascript
onConnect: function(status) {
    // webRTC receives a call connection event (incoming or outgoing).
    if (status == SNC.Notify.Status.OPEN) {
        setStatus(getTimeStamp() + " -- Successfully established call");
        showHangupButton(); // update the UI
    }
},
```

**onDisconnect**

The `onDisconnect` function runs when a call disconnects.

**onIncoming**

The `onIncoming` function runs when a call is made to the client and the telephony provider returns the parent call ID.

This function exposes these parameters as a single JSON object. Use these parameters when implementing the event handler function.

**Table 860: Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callInfo.from</td>
<td>String</td>
<td>The caller’s phone number.</td>
</tr>
<tr>
<td>callInfo.to</td>
<td>String</td>
<td>The called phone number.</td>
</tr>
</tbody>
</table>
### onIncoming

The `onIncoming` function runs when an incoming call is made from the client. This function exposes one parameter. Use this parameter when implementing the event handler function.

```javascript
onIncoming: function(callInfo) {
  gs.log('incomming call from : ' + callInfo.from);
  var gru = new GlideRecord('sys_user');
  gru.addQuery('phone', callInfo.from);
  gru.query(function() {
    while (gru.next()) { // there may be more than one person with same number?
      // fetch caller's info
      gs.log('caller : ' + gru.name);
    }
  });
},
```

### Table 861: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callInfo.callSid</td>
<td>String</td>
<td>The SID of the call from the telephony provider.</td>
</tr>
</tbody>
</table>

### onOutgoing

The `onOutgoing` function runs when an outgoing call is made from the client.

### onAccept

The `onAccept` function when the client accepts an incoming call.

### onMute

The `onMute` function runs when the client is muted.

### onUnmute

The `onUnmute` function runs when the client is unmuted.

### onCancel

The `onCancel` function runs if the caller cancels the call.
Notify client building a configuration object example

This example demonstrates how to construct the configuration object needed to instantiate a Notify WebRTC Client object.

The configuration object is a JSON object that specifies properties such as the telephony service provider and caller information, and event handler functions.

This sample code demonstrates how to create a simple configuration object.

```javascript
var notifyConfig = {
    vendor: SNC.Notify.Vendor.TWILIO, //Pass one of the supported vendor constants. Supported value is SNC.Notify.Vendor.TWILIO. Do not pass raw string values.
    callerId: '19991231234', //Pass a valid phone number to use as the caller ID. This number is usually provided by your telephony vendor.
    autoReconnect: true, //Pass true to automatically create a new WebRTC vendor session when the current session expires.
    onReady: function() {},
    onOffline: function() {},
    onError: function( message ) {},
    onConnect: function( status ) {},
    onDisconnect: function() {},
    onIncoming: function( {from, to, callSid} ) {},
    onOutgoing: function( callSid ) {},
    onAccept: function() {},
    onMute: function() {},
    onUnmute: function() {},
    onCancel: function() {}
};
```

Activate Notify

The Notify plugin (com.snc.notify) requires a separate subscription. This plugin includes demo data and activates related plugins if they are not already active.

Role required: admin

Notify activates these related plugins if they are not already active.

Table 862: Plugins for Notify

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E164 Compliant Phone Number</td>
<td>Provide E-164 compliant phone number support.</td>
</tr>
<tr>
<td>[com.glide.phone_number]</td>
<td></td>
</tr>
</tbody>
</table>

To purchase a subscription, contact your ServiceNow account manager. After purchasing the subscription, activate the plugin within the production instance.

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.
   
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the Load demo data check box.
   
   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

**Migrating from legacy Notify**

When migrating to Notify from the legacy Notify functionality, several changes are made to the instance.

**Automatic changes**

Several automatic changes occur when you activate Notify if the legacy Notify functionality is already enabled.

- The legacy **Notify** application menu is removed.
- Two separators are added to the new **Notify** application menu, **Legacy Notify** and **Legacy Notify Admin**. New modules that replace the legacy **Notify** modules are added to these sections.

  **Note:** The added modules are not the same modules that were in the legacy **Notify** application menu. Any customizations made to the legacy Notify modules are not preserved.

- The notifynow_admin role allows access to Notify modules.
- The notify_admin role allows access to legacy Notify modules.
- All workflow activities from the legacy Notify application are moved to the **Legacy Notify** group.

**Legacy Notify**

Notify enables organizations with a Twilio account to send notifications using text and voice messages.

  **Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see **Notify** on page 2910.

It also allows conference calls between ServiceNow users to enable quick communications.

When Notify is active, you can configure ServiceNow to automatically generate and send notifications to selected contacts, for instance when a new incident alert is raised in the incident alert management process.

Users with the notifynow_admin role can set properties and monitor message and conference call activities.

Notify has been implemented for use within incident alert management. Refer to the Notify API documentation for details on how to implement Notify for use within other ServiceNow applications.

**Working with Legacy Notify**

Follow this process to enable and use Notify.

  **Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see **Notify** on page 2910.

1. Set up a Twilio account to provide phone connectivity for Notify.
2. Activate Notify.
3. Configure Notify to use Twilio.
4. Begin using Notify immediately within incident alert management.
5. Use the Notify API with other ServiceNow applications to provide SMS, voicemail, email, and conference calls for those applications.

View a Legacy Notify question

Questions contain a response action and a set of response choices. Use these questions to simplify communications. Then, you can monitor the resulting communication thread.

Role required: notifynow_admin

---

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

For example, the process can be used to ask members of a specific on-call group to acknowledge or reject an incident. Depending on the answer from the first team member, the incident is assigned to that person or an SMS question is sent to the next on-call member. In this example, the questions are used as part of a workflow.

To see a list of predefined questions both Notify and On-Call Scheduling need to be active.

1. Navigate to Notify > Questions.
2. Click the question **On-Call responsibility Accept/Reject** to see the question details.
3. Click the **Notify Response Choices** related list to view or edit the choices. The person who receives the SMS question can either choose to accept or reject the assignment. If the assignment is accepted, the **Response action** can be that the incident is actually assigned to that person. The response action is a script that you can modify or replace. If the assignment is rejected, an SMS question may be sent to the next person in the escalation chain, and so on. You can combine questions with a workflow to automate a process like on-call scheduling escalation.

**Create a question in legacy Notify**
Define a question that can be sent to users.

**Role required:** notifynow_admin

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see *Notify* on page 2910.

1. Navigate to **Notify > Questions**.
2. Click **New**.
3. Fill in the fields, as appropriate.

### Table 863: Notify New Question form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a description of the question.</td>
</tr>
<tr>
<td>Question</td>
<td>Enter the question you want to send. You can enter variables in the question with the <code>{ }</code> tags, for example, <code>{0}</code>. When using more than one variable, use consecutive numbering. When using multiple variables, reference them by specifying their position in the parameter list, starting from <code>{0}</code>. The variable is picked up by the script and translated into, for example, an incident or a problem number.</td>
</tr>
<tr>
<td>Params</td>
<td>Enter parameters which map to the fields on the record that the workflow was triggered for. This enables the message to show dynamic data. Multiple parameters should be separated by a comma. For example, number, shows the incident number in case the workflow was triggered from an incident. A second parameter could be, for example, short_description.</td>
</tr>
<tr>
<td>Response Action</td>
<td>Select the action that must be taken. This is a script that can be modified, or you can create a new script by clicking the search button and selecting <strong>New</strong>. Several examples of scripts are shown. Write your own script and click <strong>Submit</strong>.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.
Create a new response choice in legacy Notify
Create a new response choice to allow users to select that response for a Notify question.

Role required: notifynow_admin

Note: This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

1. Navigate to Notify > Questions.
2. Open the notify question you have just created.
3. Click New to create a new notify response choice.
4. Fill in the fields, as appropriate.

Table 864: Notify New Response Choice form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>The abbreviation for the response choice in the text message. For example, ACC for accept or REJ for reject.</td>
</tr>
<tr>
<td>Text</td>
<td>The text displayed for this response choice.</td>
</tr>
<tr>
<td>Order</td>
<td>A number indicating the order in the list of response choices. Choices with lower order numbers are listed before choices with higher order numbers.</td>
</tr>
</tbody>
</table>

5. Click Submit.

View Legacy Notify messages

Applications implementing Notify, such as incident alert management, can send Short Message Service (SMS) text messages to relevant contacts under predefined conditions, such as when a new incident alert has been created.

Role required: notifynow_admin

Note: This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

1. Navigate to Notify > Messages.
2. Click a message to see the message details.
View Legacy Notify conversations

The Notify Conversation [notifynow_conversation] table tracks bi-directional communications and adds the ability to have more than one active conversation with the same phone number at the same time.

Role required: notifynow_admin

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see [Notify](#) on page 2910.

The thread number is combined with the phone number to link the SMS or email response to the correct conversation. Conversations remain active until a response is received or until they are automatically deactivated after the thread release time runs out. The default thread release time is 10 days. To change this value, add the `nn.thread.release.age` property.

1. Navigate to **Notify > Conversations.**
2. Click a conversation record to see the conversation details.

The State of the conversation can have one of the following values:
Table 865: Notify Conversation State Descriptions

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ended</td>
<td>The conversation has been completed.</td>
</tr>
<tr>
<td>Failed</td>
<td>The conversation has failed.</td>
</tr>
<tr>
<td>Pending</td>
<td>The SMS or email is waiting to be sent.</td>
</tr>
<tr>
<td>Sent</td>
<td>The SMS or email has been sent.</td>
</tr>
<tr>
<td>Unanswered</td>
<td>The SMS or email has not been answered.</td>
</tr>
</tbody>
</table>

**Note:** The mode of conversation cannot be changed during one conversation thread. For example, you cannot switch between SMS and email during one conversation.

View Legacy Notify conference calls

Applications that use Notify, such as incident alert management, can launch and maintain conference calls between involved parties.

Role required: notifynow_admin

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

1. Navigate to **Notify > Conference Calls**.
   
   You can also access conference call information from the relevant record, such as the **Conference Calls** related list in an incident alert record.

2. Click a conference call in the list to view details.
3. Click the arrow beside the check box to expand a Participant entry. This shows participant session information, including details of the conference calls that the participant has been involved in.

4. Click a Participant name to see more detailed information about that participant.
Legacy public conference calls
A Notify conference call can be public.

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

The **Private** field on the conference call record indicates if a call is private (selected) or public (not selected). By default, all calls created automatically, such as those created through Incident Alert Management, are public. You can create private calls using the Notify API `initiateConferenceCall` method.

When created, public conference calls generate an associated **Code** which is sent via SMS to all invited participants. These participants can distribute the code to allow other users to join the public call. An administrator can control the format of the code using the property `glide.notifynow.conference_call.code.pattern`. Anyone with the code can connect to a public conference call using one of these methods:

- By calling the Twilio phone number and entering the code for that conference call.
- By sending the code in an SMS message to the Twilio phone number. Participants that join a public call this way are considered ad-hoc participants, indicated on the participant record.

Administering Legacy Notify
An administrator can set up Notify, manage Notify properties and conference call participants.

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.
Configuring legacy Notify to use the Twilio service

Each ServiceNow instance using Notify requires a valid Twilio account and telephone number. Refer to the Twilio documentation for detailed instructions.

Note: This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

After the Twilio account is set up, perform the following to use that account with Notify:

• Associate the account with Notify
• Configure Twilio with ServiceNow endpoints

Associate a Twilio account in legacy Notify
Associate a Twilio account with your instance to use that account for legacy Notify.

Role required: notifynow_admin

Note: This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

1. Navigate to Notify > Administration > Properties.
2. Enter the **AccountSID**, **AuthToken** and **phone number** values. These values can be obtained from the Twilio dashboard:

When the Twilio account details are entered in the **Notify Properties** page, the account status is updated.
3. [Optional] To use Notify on multiple ServiceNow instances, activate Notify on each instance and create a separate Twilio account and telephone number for each instance.

**Note:** Some telephone numbers are voice capable, but not SMS capable. This is shown in the Notify properties with relevant messages. Two methods are available to check this: `isSMSCapable` and `isVoiceCapable`.

---

**Configuring the Twilio service with endpoints in legacy Notify**

In order for the Twilio service to receive Notify commands, the Request URL field value for every Twilio telephone number must point to the ServiceNow instance that uses Notify with that Twilio telephone number.

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

To set these values, do one of the following:

- Define them manually in the Twilio service dashboard.
- Ensure the fields are blank in the Twilio service dashboard, then open the Notify Properties page. Notify automatically configures the correct endpoints if the Request URL fields are blank.

Set these values from within the Twilio service dashboard.
Figure 794: Twilio Numbers Dashboard

The notifyusa values underlined in the image should be replaced with your ServiceNow instance name, for:

- Voice Request URL: https://notifyusa.service-now.com/NotifyNowCallProcessor.do

© 2017 ServiceNow. All rights reserved.
• Status Callback URL: https://notifyusa.service-now.com/NotifyNowCallStatusProcessor.do
• Messaging Request URL: https://notifyusa.service-now.com/NotifyNowSmsProcessor.do

View and edit legacy Notify properties

You can view and edit Notify properties.

Role required: notifynow_admin

**Note:** This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

1. Navigate to Notify > Administration > Properties.
2. Fill in the fields.

Table 866: Notify Properties form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Status</td>
<td>Information showing the status of the associated Twilio account.</td>
</tr>
<tr>
<td>Twilio AccountSID</td>
<td>The Twilio account AccountSID, acting as the user name for that account.</td>
</tr>
<tr>
<td>Twilio AuthToken</td>
<td>The Twilio account AuthToken, acting as the password for that account. See Associating a Twilio Account.</td>
</tr>
<tr>
<td>Twilio SMS and voice enabled phone number</td>
<td>The Twilio number that Notify will use. Telephone numbers must be entered in the E.164 format and need to exist under the Twilio account. See Associating a Twilio Account.</td>
</tr>
<tr>
<td>Number of frequent conference call participants to be displayed</td>
<td>The number of people to display in the frequently called list.</td>
</tr>
<tr>
<td>Voice</td>
<td>The voice used for Notify communications. Select <strong>woman</strong> or <strong>man</strong> to provide female or male voice support for English, Spanish, French, German, and Italian. Select <strong>alice</strong> to provide female voice support for a wider range of languages.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.

*Legacy Notify account status messages*

Account status messages are visible on the Notify Properties page.

Table 867: Account Status Message Descriptions

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your account is ready for use</td>
<td>Notify is correctly configured and ready for use.</td>
</tr>
<tr>
<td>Your Twilio AccountSID or AuthToken are not valid</td>
<td>An incorrect value has been entered in the AccountSID or AuthToken fields in the Notify properties.</td>
</tr>
<tr>
<td>Your Twilio phone number is not valid</td>
<td>Incorrect telephone number information is defined in the Notify properties.</td>
</tr>
<tr>
<td>Your Twilio phone number does not have properly configured endpoints</td>
<td>The Request URL endpoint settings have not been properly configured.</td>
</tr>
<tr>
<td>Your Twilio account is not configured properly</td>
<td>One or more of your Twilio account settings is incorrect. Open the Twilio dashboard and check the settings.</td>
</tr>
</tbody>
</table>
Manage legacy Notify conference call participants

As a Notify administrator, you can mute, unmute, and kick participants on a conference call.

Role required: notifynow_admin

### Note:
This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

Muting a participant prevents that person from speaking on the conference call but does not notify any participants, including the muted participant. Kicking a participant removes that person from the conference call and sets the Response field for the participant session to kicked.

1. Navigate to Notify > Conference Calls.
2. Select an active conference call.
3. In the Notify Conference Call Participants related list, select a participant.
4. Click the button for the action you want to perform, such as Mute to mute the participant.

You can perform these actions for multiple participants as a single operation using the Actions on selected rows choice list on the Notify Conference Call Participants related list.

Installed with Legacy Notify

Activating the Notify plugin adds or modifies tables, user roles, script includes, and other components.

### Note:
This content applies only to the Legacy Notify application, available prior to the Geneva release. For information about the Notify application available starting with Geneva, see Notify on page 2910.

### Tables

Notify adds or modifies the following tables.

#### Table 868: Tables

<table>
<thead>
<tr>
<th>Display Name [Table Name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer [notifynow_answer]</td>
<td>Information for possible answers to the SMS question sent.</td>
</tr>
<tr>
<td>Conversation [notifynow_conversation]</td>
<td>Information for the conversation thread.</td>
</tr>
<tr>
<td>NotifyNow Conference Call [notifynow_conference_call]</td>
<td>Information for conference call records.</td>
</tr>
<tr>
<td>NotifyNow Participant Session [notifynow_participant_session]</td>
<td>Information regarding conference call sessions for individual conference call participants.</td>
</tr>
<tr>
<td>Participant [notifynow_participant]</td>
<td>Information regarding individual conference call participants.</td>
</tr>
<tr>
<td>Question [notifynow_question]</td>
<td>Information for the possible questions in the SMS.</td>
</tr>
</tbody>
</table>
### Display Name [Table Name]

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response_action [notifynow_response_action]</td>
<td>Information for the notify response action associated with the question.</td>
</tr>
<tr>
<td>Response_choice [notifynow_response_choice]</td>
<td>Information for the notify response choice associated with the question.</td>
</tr>
<tr>
<td>SMS Messages [notifynow_message]</td>
<td>Information for the actual SMS messages sent.</td>
</tr>
</tbody>
</table>

## Properties

Notify adds the following system properties.

### Table 869: Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.notifynow.frequent_participant_count</td>
<td>The number of frequent conference call participants to be displayed.</td>
</tr>
<tr>
<td>glide.notifynow.voice</td>
<td>The voice used for Notify communications.</td>
</tr>
<tr>
<td>glide.notifynow.twilio.accountsid</td>
<td>The Twilio account AccountSID.</td>
</tr>
<tr>
<td>glide.notifynow.twilio.token</td>
<td>The Twilio account AuthToken.</td>
</tr>
<tr>
<td>glide.notifynow.twilio.phonenumber</td>
<td>The Twilio number that Notify will use.</td>
</tr>
<tr>
<td>glide.notifynow.twilio.answering_machine_detection</td>
<td>A boolean flag to enable or disable answering machine detection. Set to true to use answering machine detection. Set to false to continue the call.</td>
</tr>
<tr>
<td>nn.thread.release.age</td>
<td>The number of days after which the conversation thread release time runs out. Defaults to 10 days.</td>
</tr>
<tr>
<td>glide.notifynow.conference_call.code.pattern</td>
<td>The pattern used to generate a conference call code for public conference calls. Number signs (#) in the pattern are replaced with random numbers when a code is generated.</td>
</tr>
<tr>
<td>glide.notifynow.fix_invalid_phone_number</td>
<td>A boolean flag to enable or disable automatic correction of invalid phone numbers. Twilio might incorrectly prefix non-US phone numbers with 1. When this property is true, the instance automatically removes the 1 for non-US calls. When this property is false, you might be incorrectly identified when reconnecting to a call using SMS.</td>
</tr>
</tbody>
</table>

## User Roles

Notify adds the following user roles.
Table 870: User Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Contains Roles</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifynow_admin</td>
<td>None</td>
<td>Administrator with privileges for Notify functionality.</td>
</tr>
</tbody>
</table>

UI Actions

Notify adds the following UI actions.

Table 871: UI Actions

<table>
<thead>
<tr>
<th>UI Action</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate Conference Call</td>
<td>Incident Alert [incident_alert]</td>
<td>Adds a link to the Incident Alert form, if that plugin is activated, which displays a dialog box for starting a conference call with selected participants.</td>
</tr>
<tr>
<td>Invite to Conference Call</td>
<td>NotifyNow Conference Call [notifynow_conference_call]</td>
<td>Adds a link to the Notify Conference Call form which displays a dialog box for adding more users to an active conference call.</td>
</tr>
</tbody>
</table>

Script Includes

Notify adds the following script includes.

Table 872: Script Includes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IncidentAlertConferenceCall</td>
<td>A utility JavaScript Prototype class allowing users to initiate the conference call and add other users to the conference call. Should be used in conjunction with the slushbucket UI page (notifynow_participant).</td>
</tr>
</tbody>
</table>

Business Rules

Notify adds the following business rules.
### Table 873: Business Rules

<table>
<thead>
<tr>
<th>Business Rule Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS on new Incident Alert</td>
<td>Incident Alert [incident_alert]</td>
<td>Sends an SMS to any default contacts added when an incident alert record is created.</td>
</tr>
<tr>
<td>Conference Call Allowed</td>
<td>Incident Alert [incident_alert]</td>
<td>Displays or hides the initiate conference call UI action by storing true or false in g_scratchpad.conferenceCallAllowed.</td>
</tr>
<tr>
<td>Update Conference Call Started IA Activity</td>
<td>NotifyNow Conference Call [notifynow_conference_call]</td>
<td>Logs when a conference call started by writing to an incident alert's comment field if the source record is from the incident_alert table.</td>
</tr>
<tr>
<td>Update Conference Call Finished IA Activity</td>
<td>NotifyNow Conference Call [notifynow_conference_call]</td>
<td>Logs when a conference call ended and what actions the conference call participants took by writing to an incident alert's comment field if the source record is from the incident_alert table.</td>
</tr>
</tbody>
</table>

### Workflow Activities

Notify adds the following workflow activities.

### Table 874: Workflow Activities

<table>
<thead>
<tr>
<th>Activity Name</th>
<th>Activity Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send Notify SMS</td>
<td>Notify</td>
<td>Sends an SMS message (maximum 1600 Characters) to an E.164 compliant mobile phone number of the selected recipients and/or groups.</td>
</tr>
<tr>
<td>Activity Name</td>
<td>Activity Category</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Send Notify SMS Question</td>
<td>Notify</td>
<td>Sends predefined SMS Question message (maximum 1600 Characters) to an E.164 compliant mobile phone number of the selected recipients and/or groups. This is one of two main activities for workflow On-Call Assign by Acknowledgement. In a predefined message, the recipient is asked to assign himself to a newly created incident. The same message also contains predefined answers.</td>
</tr>
<tr>
<td>Send Email Question</td>
<td>Notify</td>
<td>Sends email generated from email template containing notification that recipient is the current On-call resource for a newly created task. This is one of two main activities for the workflow On-Call Assign by Acknowledgement. In a predefined message, the recipient is asked to assign himself to a newly created incident. The email contains two links that enable the user to accept or reject assignment.</td>
</tr>
</tbody>
</table>

**Legacy Notify API**

The legacy Notify API provides functionality for sending emails, sending SMS messages, and setting up conference calls.

Use this when you want to use Notify functionality with applications on your system.

**Note:** This API is included with the legacy Notify functionality. For APIs included in the current Notify feature, see the Notify, NotifyAction, NotifyPhoneNumber, and NotifyClient APIs.

**NotifyNow - getReadyState()**

Indicate whether Notify is set up correctly or not.

This method can only be accessed by administrators or users with the notifynow_admin role. Users with all other roles get the message False when trying to run the function in a script.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
### Table 876: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>True if Notify is set up correctly, otherwise false.</td>
</tr>
</tbody>
</table>

```javascript
var nn = new SNC.NotifyNow();
gs.log(((nn.getReadyState()) ? "OK" : "NOT OK"));
```

#### NotifyNow - getStatus()

Get the current status of Notify configuration.

This method can only be accessed by administrators or users with the notifynow_admin role. Users with all other roles get the message Unauthorized when trying to run the function in a script.

### Table 877: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>One of the possible status messages.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO_NUMBER_MESSAGE</td>
<td>The account does not have a telephone number set up. Ensure that you set up the telephone number for the account.</td>
</tr>
<tr>
<td>NO_ENDPOINTS_MESSAGE</td>
<td>The account does not have its endpoints set up correctly. Ensure that you set up the endpoints for the account.</td>
</tr>
<tr>
<td>ACCOUNT_OK_MESSAGE</td>
<td>The account is active and ready for use.</td>
</tr>
<tr>
<td>ACCOUNT_NO_AUTH</td>
<td>The Twilio AuthToken is not valid.</td>
</tr>
<tr>
<td>ACCOUNT_NOT_CONFIGURED</td>
<td>AccountSID or AuthToken is not valid.</td>
</tr>
</tbody>
</table>

```java
var nn = new SNC.NotifyNow();
gs.log(nn.getStatus());
```

**NotifyNow - initiateConferenceCall(String[] conferenceCallParticipants, String conferenceCallTitle)**

Initiate a new conference call.

**Table 879: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conferenceCallParticipants</td>
<td>String</td>
<td>One or more users, conference call participants, identified by the sys_ids from the sys_user table or E.164-compliant phone numbers.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>conferenceCallTitle</td>
<td>String</td>
<td>Title of the conference call. This parameter has a maximum length of 40 characters.</td>
</tr>
</tbody>
</table>

Table 880: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideRecord</td>
<td>The conference call record, or null if there was an error.</td>
</tr>
</tbody>
</table>

This initiates a conference call with E.164-compliant phone numbers for participants, without the optional source record parameter and and does not send any conference call details via SMS or email.

```javascript
var participants = ['+31205655548', '+31205655552', '+31652825393'];
// set up conference call
var nn = new SNC.NotifyNow();
var conferenceCall = nn.initiateConferenceCall(participants, "testing12");
gs.log('started conference call: ' + conferenceCall.getUniqueValue());
```

NotifyNow - initiateConferenceCall(String[] conferenceCallParticipants, String conferenceCallTitle, GlideRecord sourceRecord, Boolean private)

Initiate a new conference call.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conferenceCallParticipants</td>
<td>String</td>
<td>One or more users, conference call participants, identified by the sys_ids from the sys_user table or E.164-compliant phone numbers.</td>
</tr>
<tr>
<td>conferenceCallTitle</td>
<td>String</td>
<td>Title of the conference call. This parameter has a maximum length of 40 characters.</td>
</tr>
<tr>
<td>sourceRecord</td>
<td>GlideRecord</td>
<td>Source record to associate to the conference call such as an incident or problem number.</td>
</tr>
<tr>
<td>private</td>
<td>Boolean</td>
<td>Value to control if a conference call is private. This value defaults to false.</td>
</tr>
</tbody>
</table>
Table 882: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideRecord</td>
<td>The conference call record, or null if there was an error.</td>
</tr>
</tbody>
</table>

This initiates a conference call with participants that have a E.164-compliant phone number and participants from the sys_user table and sends conference call details via SMS and email to all participants.

```java
// define phone number participants
var participants = ['+31205655548', '+31205655552', '+31652825393'];

// we also want to add two Dutch sys_user participants
var user = new GlideRecord('sys_user');
user.addNotNullQuery('mobile_phone');
user.addQuery('mobile_phone', 'STARTSWITH', '+316');
user.setLimit(2);
user.query();

// add users to the participant array
while (user.hasNext() && user.next()) {
    gs.log('adding user ' + user.getValue('name') + ' with phone number ' +
           user.getValue('mobile_phone') + ' to the participant array');
    participants.push(user.getUniqueValue());
}

// define a source record to associate with the conference call
var source = new GlideRecord("cmdb_ci");
source.query("asset_tag", "P1000167");
if (source.hasNext() && source.next()) {
    // set up conference call
    var nn = new SNC.NotifyNow();
    var conferenceCall = nn.initiateConferenceCall(participants,
                                                   "testing 1 2", source);

    // check if the conference call was successfully created
    if (conferenceCall != null) {
        gs.log('started conference call: ' +
                conferenceCall.getUniqueValue());
    } else {
        gs.log('could not start the conference call :(');
    }
}
```

NotifyNow - isCallable(String participant)

Determine whether a user is callable or not.

A user must have a valid phone number to be callable. A user who is already in an active session is not callable.
### Table 883: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>String or GlideRecord</td>
<td>A sys_user or notifynow_participant record, or an E.164-compliant phone number.</td>
</tr>
</tbody>
</table>

### Table 884: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Whether this participant can be called or not.</td>
</tr>
</tbody>
</table>

```javascript
var nn = new SNC.NotifyNow();
gs.log('by number: ' + nn.isCallable('+31612345678'));

var user = GlideRecord('sys_user');
user.query('sys_id', '13d39544eb5201003cf587b9d106fe9');
if (user.hasNext() && user.next())
gs.log('by user: ' + nn.isCallable(user));

var participant = GlideRecord('notifynow_participant');
participant.query('sys_id', '33b11430eb1201003cf587b9d106feb9');
if (participant.hasNext() && participant.next())
gs.log('by participant: ' + nn.isCallable(participant));
```

### NotifyNow - isSMSCapable()

Check if the telephone number associated with the Twilio account is capable of sending SMS messages.

### Table 885: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 886: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Whether the telephone number associated with the Twilio account is capable of sending SMS messages.</td>
</tr>
</tbody>
</table>

```javascript
gs.log('The twilio number is SMS capable: ' + ((new SNC.NotifyNow().isSMSCapable()) ? 'yes' : 'no'));
```
NotifyNow - isSMSCapable(String userID)

Check if a user is able to send SMS messages.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userID</td>
<td>String</td>
<td>The sys_id of the user you want to check for an SMS-capable phone number.</td>
</tr>
</tbody>
</table>

Table 888: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>If the user can send SMS messages.</td>
</tr>
</tbody>
</table>

```javascript
gs.log('the user is able to send SMS messages (e.g. has a SMS device): ' + ((new SNC.NotifyNow().isSMSCapable('&lt;user sys_id&gt;')) ? 'yes' : 'no'));
```

NotifyNow - isVoiceCapable()

Check if the telephone number associated with the Twilio account is capable of setting up phone calls.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 890: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Whether the telephone number associated with the Twilio account is capable of setting up phone calls.</td>
</tr>
</tbody>
</table>

```javascript
gs.log('the Twilio number is Voice capable: ' + ((new SNC.NotifyNow().isVoiceCapable()) ? 'yes' : 'no'));
```

NotifyNow - isVoiceCapable(String userID)

Check if a user is able to make voice calls.
Table 891: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userID</td>
<td>String</td>
<td>The sys_id of the user you want to check for a voice-call capable phone number.</td>
</tr>
</tbody>
</table>

Table 892: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Whether the user has a voice-call capable phone number.</td>
</tr>
</tbody>
</table>

gs.log('the user is able to send SMS messages (e.g. has a SMS device): ' +
        ((new SNC.NotifyNow().isVoiceCapable('someuserid')) ?
         'yes' : 'no'));

NotifyNow - kick(GlideRecord participant)
Remove a participant from a conference call.

Table 893: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>GlideRecord</td>
<td>The conference call participant to remove from the call.</td>
</tr>
</tbody>
</table>

Table 894: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>True if the participant was removed, otherwise false.</td>
</tr>
</tbody>
</table>

var participantId = "<participant sys_id>";
var participant = new GlideRecord('notifynow_participant');
participant.get(participantId);
if (participant.isValid()) {
    // kick participant
    result = new SNC.NotifyNow().kick(participant);
    gs.log('participant kicked: ' + result);
}
NotifyNow - mute(GlideRecord participant)

Mute a participant on a conference call.

Table 895: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>GlideRecord</td>
<td>The conference call participant to mute.</td>
</tr>
</tbody>
</table>

Table 896: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>True if the participant was muted, otherwise false.</td>
</tr>
</tbody>
</table>

```
var participantId = "<participant sys_id>";
var participant = new GlideRecord('notifynow_participant');
participant.get(participantId);
if (participant.isValid()) {
    // mute participant
    result = new SNC.NotifyNow().mute(participant);
    gs.log('participant muted: ' + result);
}
```

NotifyNow - umute(GlideRecord participant)

Unmute a participant on a conference call.

Table 897: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>GlideRecord</td>
<td>The muted conference call participant to unmute.</td>
</tr>
</tbody>
</table>

Table 898: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>True if the participant was unmuted, otherwise false.</td>
</tr>
</tbody>
</table>

```
var participantId = "<participant sys_id>";
var participant = new GlideRecord('notifynow_participant');
paticipant.get(participantId);
if (participant.isValid()) {
```

© 2017 ServiceNow. All rights reserved. 2987
// unmute participant
result = new SNC.NotifyNow().unmute(participant);
gs.log('participant unmuted: ' + result);  
}

NotifyNow - sendEmailQuestion(String emailAddress, String question, GlideRecord sourceRecord, String emailSubject)

Send an email question to an email address.
The sendEmailQuestion method produces a question body and requires users to click a link to indicate their choice.

Table 899: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emailAddress</td>
<td>String</td>
<td>Email address to send the question to.</td>
</tr>
<tr>
<td>question</td>
<td>String or GlideRecord</td>
<td>The question record to send or the sys_id of a question record.</td>
</tr>
<tr>
<td>sourceRecord</td>
<td>GlideRecord</td>
<td>An optional source record to associate to the SMS question, such as an incident.</td>
</tr>
<tr>
<td>emailSubject</td>
<td>String</td>
<td>Optional text to override the default email subject.</td>
</tr>
</tbody>
</table>

Table 900: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The conversation sys_id.</td>
</tr>
</tbody>
</table>

This example demonstrates using the default email subject.

```javascript
var user = GlideRecord("sys_user");
user.get("email", "someone@somedomain.com");

new SNC.NotifyNow().sendEmailQuestion(user.getValue('email'),
  "b6b34500bf3111003cf5585ce2c0739ce", user);
```

This example uses dot-walking and specifies a source record and email subject.

```javascript
new SNC.NotifyNow().sendEmailQuestion("someone@somedomain.com",
  "b6071733bf1111003cf585ce2c07390f", current,
  "Please answer this question");
```
This example uses dot-walking and specifies an email subject but no source record.

```java
new SNC.NotifyNow().sendEmailQuestion("someone@somedomain.com", 
    "b6071733bf1111003cf585ce2c07390f", 
    "Please answer this question");
```

**NotifyNow - sendSMS(String phoneNumber, String smsBody)**

Sends an SMS message to an E.164-compliant mobile phone number.
Notify supports international numbers. Using this method with a number that does not support sending SMS messages results in an error being logged.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phoneNumber</td>
<td>String</td>
<td>The E.164-compliant phone number to send the message to.</td>
</tr>
<tr>
<td>smsBody</td>
<td>String</td>
<td>The message to send, maximum 1600 characters.</td>
</tr>
</tbody>
</table>

**Table 902: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
new SNC.NotifyNow().sendSMS("+31612345678", "This is a message without source record");
```

**NotifyNow - sendSMS(String phoneNumber, String smsBody, GlideRecord source)**

Send an SMS message to an E.164-compliant mobile phone number.
Notify supports international numbers. Using this method with a number that does not support sending SMS messages results in an error being logged.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phoneNumber</td>
<td>String</td>
<td>The E.164-compliant phone number to send the message to.</td>
</tr>
<tr>
<td>smsBody</td>
<td>String</td>
<td>The message to send, maximum 1600 characters.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>source</td>
<td>GlideRecord</td>
<td>The source record to associate with this SMS message.</td>
</tr>
</tbody>
</table>

Table 904: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var source = new GlideRecord("my_table");
source.query("my_field", "my_value");

if (source.hasNext() && source.next()) {
    // send a text message
    var nn = new SNC.NotifyNow();
    var message = "this is just a test";
    var number = "+31612345678";
    nn.sendSMS(number, message, source);
}
```

This example uses dot-walking and the current record as the source record.

```javascript
new SNC.NotifyNow().sendSMS("+31612345678", "this is a test", current);
```

NotifyNow - sendSMSQuestion(String phoneNumber, String question, GlideRecord sourceRecord)

Send an SMS question.

Table 905: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phoneNumber</td>
<td>An E.164-compliant phone number to send the message to.</td>
<td></td>
</tr>
<tr>
<td>question</td>
<td>The question record to send or the sys_id of a question record.</td>
<td></td>
</tr>
<tr>
<td>sourceRecord</td>
<td>An optional source record to associate to the SMS question, such as an incident.</td>
<td></td>
</tr>
</tbody>
</table>
Table 906: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The conversation sys_id, or null if the SMS was not sent successfully.</td>
</tr>
</tbody>
</table>

```
var question = new GlideRecord("notifynow_question");
question.query();

// get the first question
if (question.hasNext() && question.next()) {
    // send the sms question
    var number = "+31612345678";
    var nn = new SNC.NotifyNow();
    nn.sendSMSQuestion(number, question.getUniqueValue(), current);
}
```

NotifyNow - addConferenceCallParticipant(String conferenceCall, String participant)

Add ad-hoc users to an ongoing conference call.

When the method is called with a phone number for the `participant` parameter and there is exactly one `sys_user` record that matches the phone number, that `sys_user` record will be related to the participant. The participant's phone number field will be left blank because the phone number is in the `sys_user` record. If there are several `sys_user` records that match the phone number, or if there are no results, the participant's phone number field will be filled in, and there will be no stored reference to `sys_user` because the user is not known.

Table 907: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conferenceCall</td>
<td>String or GlideRecord</td>
<td>The sys_id or GlideRecord of an active conference call.</td>
</tr>
<tr>
<td>participant</td>
<td>String or GlideRecord</td>
<td>The sys_id or GlideRecord of a user with an E.164-compliant phone number, or an E.164-compliant phone number.</td>
</tr>
</tbody>
</table>

Table 908: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideRecord</td>
<td>The participant record of the new participant that was added to the conference call.</td>
</tr>
</tbody>
</table>

```
// add a new participant by conference call sys_id (string) and phone number (string)
```
```javascript
var nn = new SNC.NotifyNow();
gs.log(nn.addConferenceCallParticipant('d193b242eb020100a04d4910f206fe39', '+31612345678'));

// add a new participant by conference call sys_id (string) and user record (GlideRecord)
var user = new GlideRecord('sys_user');
user.query('user_name', 'myUserName');
if (user.hasNext() && user.next()) {
    var nn = new SNC.NotifyNow();
    gs.log(nn.addConferenceCallParticipant('d193b242eb020100a04d4910f206fe39', user));
    // you could have added the user by sys_id as well:
    // nn.addConferenceCallParticipant('d193b242eb020100a04d4910f206fe39', user.getValue('sys_id'));
} else {
    gs.log('no such user');
}

// add a new participant by conference call record (GlideRecord) and phone number (string)
var conferenceCall = new GlideRecord('notifynow_conference_call');
conferenceCall.query('title', 'IA0001001');
if (conferenceCall.hasNext() && conferenceCall.next()) {
    var nn = new SNC.NotifyNow();
    gs.log(nn.addConferenceCallParticipant(conferenceCall, '+31612345678'));
} else {
    gs.log('no such conference call');
}
```

NotifyNow - convertLocalPhoneNumberToE164(String userID, String phoneNumber)

Convert a local phone number to an E.164-compliant phone number based on a user's location.

**Table 909: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userID</td>
<td>String</td>
<td>The sys_id of a sys_user record to get location information from.</td>
</tr>
<tr>
<td>phoneNumber</td>
<td>String</td>
<td>The phone number.</td>
</tr>
</tbody>
</table>
Table 910: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The E.164-compliant phone number.</td>
</tr>
</tbody>
</table>

```javascript
var localPhoneNumber = '01784 221600';
var userName = 'Heath Vanalphen';

var user = new GlideRecord('sys_user');
user.get('name', userName);
var E164Number = new SNC.NotifyNow().convertLocalPhoneNumberToE164(user.getUniqueValue(),
localPhoneNumber);
gs.log('converted: ' + localPhoneNumber + ' to ' + E164Number + '
  based on ' + user.getValue('name') + '
  \'s location (' + user.getValue('location') + ')');
```

**NotifyNow - getConferenceCallParticipants(String conferenceCallId, Boolean isCallable)**

Get all participants for a conference call.

Table 911: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conferenceCallId</td>
<td>String</td>
<td>The ID of the conference call.</td>
</tr>
<tr>
<td>isCallable</td>
<td>Boolean</td>
<td>An optional flag to return either only the users you can call (true) or those you cannot call (false).</td>
</tr>
</tbody>
</table>

Table 912: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideRecord</td>
<td>The participants</td>
</tr>
</tbody>
</table>

```javascript
var nn = new SNC.NotifyNow();
var user = 
nn.getConferenceCallParticipants('c2e91710eb120100f34087b9d106fe37');

while (user.hasNext() && user.next()) {
   if (user.getValue('participant')) {
      gs.log('user: ' + user.getValue('sys_id'));
   } else {
      gs.log('phone number: ' +
      user.getValue('phone_number'));
   }
}
var nn = new SNC.NotifyNow();
var user =
nn.getConferenceCallParticipants('c2e91710eb120100f34087b9d106fe37',
true);

while (user.hasNext() && user.next()) {
  if (user.getValue('participant')) {
    gs.log('user: ' + user.getValue('sys_id'));
  } else {
    gs.log('phone number: ' +
user.getValue('phone_number'));
  }
}

var conferenceCallId = '32b11430eb1201003cf587b9d106feb8';

// get all participants
 gs.log('all conference call participants:');
var nn = new SNC.NotifyNow();
var user = nn.getConferenceCallParticipants(conferenceCallId);
gs.log(user);

// get all callable participants
 gs.log('all conference call participants we can call:');
user = nn.getConferenceCallParticipants(conferenceCallId, true);
gs.log(user);

// get all un callable participants
 gs.log('all conference call participants that are already in an
active session and whom we cannot call:');
user = nn.getConferenceCallParticipants(conferenceCallId,
false);
gs.log(user);

### NotifyNow - getFrequentlyCalledUsers(Number limit)

Return a number of frequently-called users, up to the limit parameter, in alphabetical order.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>limit</td>
<td>Number</td>
<td>The maximum number of results.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
Table 914: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideRecord</td>
<td>The frequently called users in alphabetical order.</td>
</tr>
</tbody>
</table>

```javascript
var nn = new SNC.NotifyNow();
var fc = nn.getFrequentlyCalledUsers(10);
while (fc.hasNext() && fc.next()) {
    gs.log("got user " + fc.getValue('name') + ' - ' + fc.getValue('sys_id'));}
```

**NotifyNow - getPreferredE164VoiceNumber(GlideRecord user)**

Get a user's preferred E.164-compliant phone number for voice calls.

Table 915: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>GlideRecord or String</td>
<td>The user record or the sys_id of a user to get the E.164-compliant phone number from.</td>
</tr>
</tbody>
</table>

Table 916: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The E.164-compliant phone number or null.</td>
</tr>
</tbody>
</table>

```javascript
var userID = "<user sys_id>";
var E164Number = new SNC.NotifyNow().getPreferredE164VoiceNumber(userID);
gs.log('the preferred phone number for setting up voice calls is ' + E164Number + ' for user with id: ' + userID);
```

**NotifyNow - getPreferredE164SMSNumber(GlideRecord user)**

Get a user's preferred E.164-compliant phone number for SMS messages.

Table 917: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>GlideRecord or String</td>
<td>The user record or the sys_id of a user to get the E.164-compliant phone number from.</td>
</tr>
</tbody>
</table>
Table 918: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The E.164-compliant phone number or null.</td>
</tr>
</tbody>
</table>

```javascript
var userID = "<user sys_id>";
var E164Number = new SNC.NotifyNow().getPreferredE164SMSNumber(userID);
gs.log('the preferred phone number for sending SMS notifications is ' + E164Number + ' for user with id: ' + userID);
```

NotifyNow - getPreferredEmailAddress(GlideRecord user)

Get a user's preferred email address

Table 919: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>GlideRecord or String</td>
<td>The user record or the sys_id of a user to get the email address from.</td>
</tr>
</tbody>
</table>

Table 920: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The email address or null.</td>
</tr>
</tbody>
</table>

```javascript
var userID = "some user sys id";
var email = new SNC.NotifyNow().getPreferredEmailAddress(userID);
gs.log('the preferred email address for sending email notifications is ' + email + ' for user with id: ' + userID);
```

Events

Event are special records the system uses to log when certain conditions occur and to take some kind of action in response to the conditions.

The system uses business rules to monitor for system conditions and to generate event records in the Event [sysevent] table, which is also known as the event log or event queue.

Event-generating business rules typically use this script logic:

If [some condition is true for the current record], then [add a specific event to the queue].

For example, here are some of the conditions in the incident event business rule:
• If a user adds a comment to an incident record, add an incident.commented event.
• If a user adds an incident record, add an incident.inserted event.
• If a user updates an incident record, add an incident.updated event.

Event-generating business rules use the GlideSystem `eventQueue` method to insert event records, which typically contain this information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The unique name of event. Baseline event names include the record effected and the triggering action such as incident.commented.</td>
</tr>
<tr>
<td>Parm1</td>
<td>An event-specific parameter the system uses to pass record information to other parts of the system, such as a record Sys ID or a field value.</td>
</tr>
<tr>
<td>Parm2</td>
<td>Another event-specific parameter the system uses to pass record information to other parts of the system, such as a record Sys ID or a field value.</td>
</tr>
<tr>
<td>Table</td>
<td>The table to which the event applies. This is the same table on which the business rule ran.</td>
</tr>
<tr>
<td>Instance</td>
<td>The Sys ID of the record to which this event applies.</td>
</tr>
</tbody>
</table>

Scheduled jobs periodically read the event queue and forward them to the appropriate handler for processing. The handler uses information from event records to take some kind of action such as:

• Send a notification
• Run a script action
• Trigger a workflow activity

By default, the system provides events covering a broad view of application activity. If existing events do not meet your needs, you can create your own events to watch for specific changes to records.

**Event registry**

The events registry lists the events the system recognizes. Use registered events to automate other activities, such as script actions or notifications.

Events can be used to schedule actions or tasks to occur when conditions are fulfilled.

Examples:

• `kb.view` - an event triggered when a user views a Knowledge Base article, used to trigger the script action Knowledge View to create a Knowledge Use record every time an article is viewed.
• `incident.commented` - an event triggered when a user comments on an article, used by two incident commented email notifications.

Inactivity Monitors:

An inactivity monitor triggers an event if a record has not been updated for a defined length of time.
Script Actions:
Script actions are scripts which are triggered when an event is recorded in the log. In that way, scripts can be set to be performed whenever a particular activity occurs in the platform, rather than at a particular time (like scheduled jobs) or in response to particular conditions (like business rules).

Notifications:
Events are also used to trigger Email Notifications when an event is recorded in the log.

Event states
The event state describes where in the life cycle the event is.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready</td>
<td>The system created the event and it is in the queue waiting to be processed.</td>
</tr>
<tr>
<td>Processed</td>
<td>The event successfully ran. An event does not necessarily trigger any further action when processed. Additional functionality must make use of the event.</td>
</tr>
<tr>
<td>Error</td>
<td>The event encountered an error during processing. This state is often caused by invalid event parameters. Reprocessing the event may resolve the error.</td>
</tr>
<tr>
<td>Transferred</td>
<td>The event was rotated to a different shard of the Event [sysevent] table. When an event is rotated, a duplicate record is created in an active shard to be processed. A scheduled job processes the event when it is next in the queue, but it is not possible to predict when this will happen as because several events may need to be processed before it. Therefore, you can reprocess the event. See Reprocess an event on page 3008.</td>
</tr>
</tbody>
</table>

Event logs
The event log records all system events that occur within the system.
This log provides the following information for all events that occur:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Date and time of the event for the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the event as listed in the Event Registry.</td>
</tr>
<tr>
<td>Parm1</td>
<td>Event-specific value that depends on the event and the recipient.</td>
</tr>
<tr>
<td>Parm2</td>
<td>Event-specific value that depends on the event and the recipient.</td>
</tr>
<tr>
<td>Table</td>
<td>Database table acted on for this event.</td>
</tr>
<tr>
<td>Processed</td>
<td>Date and time the event was processed This time reflects the locale of the machine running the instance.</td>
</tr>
<tr>
<td>Processing time</td>
<td>Time taken to process this event, in milliseconds.</td>
</tr>
<tr>
<td>Queue</td>
<td>Processor queue name.</td>
</tr>
</tbody>
</table>

The incident events business rule

The incident events business rule comes with the system and defines a number of events that can be triggered by different actions in the Incident table.
This business rule defines several events, three of which are triggered after a record in the Incident table is inserted or updated. The first script is:

```java
if (current.operation() != 'insert' && current.comments.changes()) {
    gs.eventQueue("incident.commented", current, gs.getUserID(), gs.getUserName());
}
```

The condition in this script requires that a change be made to the Comments field in an existing (not inserted) incident record. If this condition is true, then the platform adds the incident.commented event to the event queue.

The second condition requires that a record be inserted before the event is added to the queue.

```java
if (current.operation() == 'insert') {
    gs.eventQueue("incident.inserted", current, gs.getUserID(), gs.getUserName());
}
```

```java
if (current.operation() == 'update') {
    gs.eventQueue("incident.updated", current, gs.getUserID(), gs.getUserName());
}
```

```java
if (!current.assigned_to.nil() && current.assigned_to.changes()) {
    gs.eventQueue("incident.assigned", current, current.assigned_to.getDisplayValue(), previous.assigned_to);
}
```

```java
if (!current.assignment_group.nil() && current.assignment_group.changes()) {
    gs.eventQueue("incident.assignedtogroup", current, current.assignment_group.getDisplayValue(), previous.assignment_group);
}
```
The third condition is true whenever the incident record is updated (including updates to the Comments field, as specified by the first script).

```java
if (current.operation() == 'update')
```

The then part of each script, the `gs.eventQueue` function, adds the event to the event queue. This statement uses the following syntax, set off with braces:

```java
gs.eventQueue("incident.updated", current, gs.getUserID(), gs.getUserName());
```

The `gs.eventQueue` function takes the following parameters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the event triggered, set in quotation marks</td>
</tr>
<tr>
<td>Record</td>
<td>The record referenced when the condition in the script evaluates to true. Usually this is expressed as current, meaning the current record the business rule is working on. If the business rule is being triggered as part of a scheduled job, use a GlideRecord argument in its place.</td>
</tr>
<tr>
<td>Parm 1</td>
<td>An optional parameter you can use to pass system or record information with the event. For example, the GlideSystem API call <code>gs.getUserID()</code> passes the Sys ID of the user who acted on the current record as a string value. Other scripts can reference this string value as parm1 using the format <code>${event.parm1}</code>.</td>
</tr>
<tr>
<td>Parm 2</td>
<td>An optional parameter you can use to pass system or record information with the event. For example, the GlideSystem API call <code>gs.getUserName()</code> passes the user name of the user who acted on the current record. Other scripts can reference this string values as parm2 using the format <code>${event.parm2}</code>.</td>
</tr>
</tbody>
</table>

Sample scripts from the change events business rule

Several scripts are found in the baseline change events business rule.

This business rule defines events that fire after a change request is inserted or updated.

```java
if (current.operation() == 'insert') {
    gs.eventQueue("change.inserted", current, gs.getUserID(), gs.getUserName());
}
if (current.operation() == 'update') {
    gs.eventQueue("change.updated", current, gs.getUserID(), gs.getUserName());
}
if (!current.assigned_to.nil() && current.assigned_to.changes()) {
```
gs.eventQueue("change.assigned", current, current.assigned_to.getDisplayValue(), previous.assigned_to.getDisplayValue());

if (current.priority.changes() && current.priority == 1) {
    gs.eventQueue("change.priority.1", current, current.priority, previous.priority);
}

if (current.risk.changes() && current.risk == 1) {
    gs.eventQueue("change.risk.1", current, current.risk, previous.risk);
}

if (current.start_date.changes() || current.end_date.changes() || current.assigned_to.changes()) {
    if (!current.start_date.nil() && !current.end_date.nil() && !current.assigned_to.nil()) {
        gs.eventQueue("change.calendar.notify", current, current.assigned_to, previous.assigned_to);
    }

    // Remove from previous assigned to, due to assigned_to changing
    if (!previous.assigned_to.nil()) {
        if (!current.assigned_to.nil() && current.assigned_to.changes()) {
            gs.eventQueue("change.calendar.notify.remove", current, current.assigned_to, previous.assigned_to);
        }
    }

    // Remove old calendar from current assigned to, due to date changing
    else if (!current.assigned_to.nil()) {
        if (current.start_date.changes() && !previous.start_date.nil()) ||
            (current.end_date.changes() && !previous.end_date.nil()) {
                gs.eventQueue("change.calendar.notify.remove", current, current.assigned_to, current.assigned_to);
            }
    }
}

Create an event

If you do not find a suitable existing event, you can create your own.

Role required: admin

1. Navigate to System Policy > Events > Registry.
2. Click New and fill in the form.
3. Click the Business Rules related link.
4. If you are creating an event for a base system table, select the existing event business rule for the table.
   For example, select the sc request events business rule to create a custom Request event.
5. If you are updating an existing event business rule, add a new condition to the Script.
   The following sample script adds a request.commented event with the user's Sys ID as parm1 and the user's user name for parm2.

```javascript
if (current.operation() != 'insert' && current.comments.changes()) {
    gs.eventQueue('request.commented', current, gs.getUserID(), gs.getUserName());
```
© 2017 ServiceNow. All rights reserved.
6. If you are creating an event for a custom table, create a new business rule that runs after database operations. For example, this business rule defines several events for a custom application called Marketing Events.

Table 924: Sample event business rule

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Attendee Events</td>
</tr>
<tr>
<td>Table</td>
<td>Attendee [x_snc_marketing_ev_attendee]</td>
</tr>
<tr>
<td>Application</td>
<td>Marketing Events</td>
</tr>
<tr>
<td>Advanced</td>
<td>Selected</td>
</tr>
<tr>
<td>When</td>
<td>after</td>
</tr>
<tr>
<td>Insert</td>
<td>Selected</td>
</tr>
<tr>
<td>Update</td>
<td>Selected</td>
</tr>
<tr>
<td>Delete</td>
<td>Selected</td>
</tr>
<tr>
<td>Script</td>
<td>Add custom script that:</td>
</tr>
<tr>
<td></td>
<td>• Checks for one or more conditions on the current record.</td>
</tr>
<tr>
<td></td>
<td>• Calls the gs.eventQueue() method and specifies an event name.</td>
</tr>
<tr>
<td></td>
<td>See code sample.</td>
</tr>
</tbody>
</table>

**Note:** If you add Filter Conditions, Role conditions, or a Condition value, verify it runs the business rule when expected.

```javascript
(function executeRule(current, previous /*null when async*/) {
    //This function will be automatically called when this rule is processed.
    if(current.operation() == 'insert' &&
        current.marketing_event.changes()) {
        gs.eventQueue('x_snc_marketing_ev.attendee.added',
                       current,
                       current.marketing_event, current.email);
    }
    //Add event when marketing event changes
    if(current.operation() == 'update' &&
        current.marketing_event.changes()) {
        gs.eventQueue('x_snc_marketing_ev.attendee.deleted',
                       previous,
                       previous.marketing_event, previous.email);
        gs.eventQueue('x_snc_marketing_ev.attendee.added',
                       current,
                       current.marketing_event, current.email);
    }
    //Add event when attendee deleted
```
7. Register the event.

Create a script action or notification to process the event.

Script actions

You can use script actions to create server-side scripts that perform a variety of tasks, such as modifying a configuration item (CI), or managing failed login attempts. Script actions are triggered by events only.

Configuration

To create a new script action, navigate to System Policy > Events > Script Actions and click New.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for your script action.</td>
</tr>
<tr>
<td>Event Name</td>
<td>Select the event to use for this script. If you do not find an event for your script action that suits your purpose, you can create a new one.</td>
</tr>
<tr>
<td>Application</td>
<td>The application that contains this script.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the script will be executed.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box (true) to enable this script action.</td>
</tr>
<tr>
<td>Condition Script</td>
<td>Create a statement for a condition under which this script should execute.</td>
</tr>
<tr>
<td>Script</td>
<td>Create a script that runs when the condition you define evaluates to true.</td>
</tr>
</tbody>
</table>

Two additional objects are available in this script:

- `event`: a GlideRecord - the sysevent that caused this to be invoked. If you wanted so get this first parameter on the event, you would use event.parm1 or event.parm2 for the second parameter. For the date/time of the event, use event.sys_created_on. To get the user ID that created the event (if there was a user associated), use event.user_id.
<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>• current: a GlideRecord - the event scheduled on behalf of (incident for example).</td>
<td></td>
</tr>
</tbody>
</table>

This is a sample of a script action that creates an email notification for Workflow activity:
Figure 796: Script action workflow

```
function sendWorkflowNotification() {
  // Get the activity that defines the information about the event
  var activity = new GlideRecord('wf_activity');
  if (activity.get(parts[0])) {
    return 0;
  }

  // The EmailAction does not know how to handle ${workflow...} constructs so we need to handle
  // these for it
  var context = new GlideRecord('wf_context');
  if (parts.length > 2) {
    context.set(parts[1]);
  }
  GlideController.putGlobal('context', context);
  var workflow = new Workflow().workflow.newWorkflowProxy();
  GlideController.putGlobal('workflow', workflow);
  var subject = workflow.activity vars subject;
  var message = workflow(activity vars .email);
  var emailAction = new GlideEmailAction();
  var email = new GlideRecord('sysevent_email_action');
```
Global events

Your instance has a global function called `global_events()` that triggers from a business rule when certain conditions occur.

This function triggers when your instance is:

- Inserting new records
- Updating existing records
- Adding comments to an existing record
- Assigning a record to a user
- Exceeding a record's inactive timer

For example, if you add the script `global.events(current)` to a business rule on the `change_request` table, the instance automatically configures the following events:

- `change_request.inserted`
- `change_request.updated`
- `change_request-commented`
- `change_request.assigned`
- `change_request.inactive`

The business rule for global events on the `change_request` table looks like this:

```
<table>
<thead>
<tr>
<th>Business Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>Table:</td>
</tr>
<tr>
<td>Order:</td>
</tr>
<tr>
<td>Active:</td>
</tr>
</tbody>
</table>

Condition:
`script`:
`global_events(current);`
```

Figure 797: Change request tasks global business rule
Reprocess an event

You can fire the event again for testing or diagnostic purposes.

Role required: admin

1. Navigate to **System Logs > Events**.
2. Open an event.
   The event returns to the event queue.

Pass event parameters from a workflow to a notification

You can pass two event parameters that send information about a record or related records from a workflow to a notification.

Role required: admin

1. Navigate to **System Policy > Events > Registry** and define a new event to call.
2. Create the activity step that calls the event from your workflow and assign the two possible parameters.
   These parameters can be references or fields on the record that triggered the workflow, such as current.number for the request item number. You can also dot-walk to records in other tables. The event then sends the parameters information to the notification that it triggers.
3. Retrieve the parameters in the notification with email scripts by using event.parm1 and event.parm2.

   For example:

   ```javascript
   var string = event.parm1.toString();
   template.print(string);
   
   var id = event.parm1.toString();
   var gr = new GlideRecord('sc_req_item');
   gr.get('sys_id', id);
   if (gr.next()){
       // Do something.
   }
   ``

Contextual search

Contextual search allows you to configure fields on forms and record producers to automatically display knowledge search results based on the text entered in those fields.

Contextual search helps users deflect or quickly resolve their issues without involving the service desk operators. Contextual search can also include results from other sources such as a service catalog, allowing a user to directly order a catalog item from a search.
Contextual searches are enabled for incidents by default, providing incident deflection using knowledge and social Q&A questions. This default implementation includes contextual search results for forms, record producers, email notifications, and wizards.

To define a contextual search, first define a search context to set parameters, then define where the contextual search results are to appear: in forms, record producers, and wizards.

You must have the admin role to define contextual searches and to administer contextual search functions and properties.

### Define contextual search

You can add and configure contextual search functions for fields in forms, record producers, and wizards.

For example, you can add a new contextual search capability to the Service Desk Call form, linking search to the short description in that form. This automatically provides relevant knowledge to service desk staff when they respond to calls, helping to resolve these calls more quickly.

By default, the system provides contextual search for incident record producers and the Incident form to help with *incident deflection and resolution*.

### Contextual search process flow

Understand the steps you follow to define the contextual search process.

**Role required:** catalog_admin or admin

1. **Define a search context to be available in multiple locations, setting overall values and conditions for the search results displayed.**
   
   For example, define a search context for service desk calls that only displays search results from the Technical Solutions knowledge base.

2. **Define where the contextual search results are to appear.**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Forms</strong></td>
<td>You can define knowledge to be associated whenever a problem record is created. For forms, you can also provide search results with email notifications sent when a record is created. For example, automatic notifications sent when an incident is created can include knowledge search results which may help the user who raised the incident resolve the issue independently.</td>
</tr>
<tr>
<td><strong>Record producers</strong></td>
<td>Define a Create New Incident record producer to trigger a search based on text entered in the Comments field. Display the search results at the bottom of the record producer.</td>
</tr>
<tr>
<td><strong>Wizards</strong></td>
<td>For example, if you have defined a wizard for creating incidents, you can add contextual search results to it.</td>
</tr>
</tbody>
</table>

- *Forms:*
- *Record producers:*
- *Wizards:*
Define search context

Create a search context to define a contextual search, with links to define which forms and record producers use that search.

For example, define a search context for service desk calls that only displays search results from your organization's Technical Solutions knowledge base.

1. Navigate to **Contextual Search > Search Contexts**.
2. Click **New**.
3. Fill in the fields, as appropriate.
Table 925: Name of form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the context.</td>
</tr>
<tr>
<td>Short description</td>
<td>Summary of the context.</td>
</tr>
<tr>
<td>Searcher</td>
<td>The searcher for this context, which determines the information source to search.</td>
</tr>
<tr>
<td></td>
<td>• Catalog only: searches service catalog items.</td>
</tr>
<tr>
<td></td>
<td>• Knowledge and catalog: searches knowledge articles and service catalog items.</td>
</tr>
<tr>
<td></td>
<td>• Knowledge and pinned knowledge: searches knowledge articles and pinned knowledge articles.</td>
</tr>
<tr>
<td></td>
<td>• Knowledge only: searches knowledge articles only.</td>
</tr>
<tr>
<td></td>
<td>• Knowledge, pinned knowledge and catalog: searches knowledge articles, pinned knowledge articles, and service catalog items.</td>
</tr>
<tr>
<td></td>
<td>• Pinned Knowledge only: searches pinned knowledge articles only.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box for activating the search context.</td>
</tr>
</tbody>
</table>

4. Right-click the form header and click **Save**.

5. Optional: Click **Set As Default** to make this the default context, used if a search query does not specify a context. For example, if a search query is invoked from a script without providing a context.

6. Use the related lists to further define the search context.

   • Record Producer Configurations: define which record producers use this context.
   • Table Configurations: define which forms use this context.
   • Wizard Configurations: define which wizards use this context.
   • Resource Configuration: define configurations to apply to the search context, such as restrictions on the results displayed.

Define contextual search for a form

Define contextual search for the form associated with a table.

You can define knowledge to be associated whenever a problem record is created. For forms, you can also provide search results with email notifications sent when a record is created. For example, automatic notifications sent when an incident is created can include knowledge search results which may help the user who raised the incident resolve the issue independently.
In the table configuration record, specify which fields on the form trigger a search. For example, a contextual search for the Incident table might display search results based on text entered in the Incident form Short description field.

1. In the Search Context form, open the Table Configurations related list and click New. Alternatively, navigate to Contextual Search > Table Configuration and click New.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Name of the table name associated with the form. For example, Incident [incident] to display search results for Incident forms.</td>
</tr>
<tr>
<td>Note: The list shows only tables and database views that are in the same scope as the contextual search.</td>
<td></td>
</tr>
<tr>
<td>Search context</td>
<td>Name of the search context this table configuration applies to.</td>
</tr>
<tr>
<td>Results header text</td>
<td>Label for the search results area that appears on the form.</td>
</tr>
<tr>
<td>Active</td>
<td>Check box for activating this context.</td>
</tr>
<tr>
<td>Results per page</td>
<td>Number of results to display per page.</td>
</tr>
<tr>
<td>Result action label</td>
<td>Label of the action button that appears when the user opens a search result record. By default, this is set to This Helped.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result action value</td>
<td>Internal value of the action button that appears when the user opens a search result record. By default, this is set to <strong>helped</strong>. This value is stored within the Relevant Document [cxs_relevant_doc] table.</td>
</tr>
<tr>
<td>Match condition</td>
<td>Conditions under which this search is enabled. For example, set <strong>[Active] [is] [true]</strong> to only enable the search on active records.</td>
</tr>
<tr>
<td>Enable search as</td>
<td>Allow you to define results to show for an additional user. See Search as different user on page 3017 for more details.</td>
</tr>
</tbody>
</table>

2. Right-click the form header and select **Save**.
3. Define which search fields to use contextual search with.
4. Optional: Define email configurations to include knowledge search results in email notifications sent.
5. Click **Update** to save and close the Table Configuration record.
6. Add the search results area to the form for the associated table.
   a) Navigate to the relevant form for that table. For example, navigate to **Incident > Open** and open an Incident form.
   b) Configure the form and add Contextual Search Results in the location where you want to display the results. For example, you might want to display the search results immediately after the text field the search is based on or at the bottom of the form.

### Define search fields on a form

Contextual searches are based on the content entered in selected search fields on forms. You can define multiple search fields for a form, with one field selected as the default.

Search results for the default field appear automatically in the Related Search Results section when you open a record. Search results in other associated fields only appear when you type content in those fields.

For example, in the base system the **Short description** is the default search field for the Incident form, so search results based on that field are displayed automatically.
### Incident - INC000055

- **Number:** INC000055
- **Caller:** Carol Coughlin
- **Location:**
- **Category:** None
- **Subcategory:** None
- **Configuration item:** SAP Sales and Distribution
- **Impact:** 1 - High
- **Urgency:** 1 - High
- **Priority:** 1 - Critical
- **Short description:** SAP Sales app is not accessible

### Related Search Results

1. **Can't access SAP**
   - SAP troubleshooting techniques Using transaction SE11, create a table (ZTABLE) with the same fields as the table in the external database, make sure that...

### Notes

- **Additional comments (Customer Notes):**

### Activity

- **2014-11-09 20:49:39** ITIL User Changed: Assigned to, Additional comments, Impact, Incident name, Opened by, Priority, Work notes Assigned to: Beth Anglin
  - **Configuration item:** SAP Sales and Distribution
  - **SAP Sales app is not accessible**

© 2017 ServiceNow. All rights reserved.
If the **Description** field is also defined as a search field on this form, search results only appear when you type content in that field.

1. Navigate to **Contextual Search > Table Configuration** and open the record.
2. Click **New** in the **Search Fields** related list.
3. Select the field to use and the order for that field.

**Note:** Only text fields can be selected.

4. Right-click the header and select **Save** to save the search field record.
5. Optional: If you have multiple search fields defined for that form, you can click **Set as Default** to select the new field as the default search field.

### Create an email notification for search results

Edit the email notification that is sent when the relevant record is created to include search results.

1. Navigate to Open the Table Configuration form for the table.
2. In the **Email Configurations** related list, click **New**.

![Image of Email Configuration form](image.png)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email notification</td>
<td>Email notification to use. For example, the default incident email notification provided is <strong>Incident opened for me</strong>.</td>
</tr>
<tr>
<td>User</td>
<td>Field identifying the user to receive the email. For example, <strong>Opened by</strong> if the email notification is sent to the user who created the record. This information is used to filter the search results included based on that user's access permissions, ensuring that the recipient can always access the information in the links provided.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Limit</td>
<td>Maximum number of results the email notification includes.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.
4. Configure the email notification to use the contextual search.
   a) Navigate to **System Policy > Email > Notifications**.
   b) Open the email notification which will include the contextual search results.
   c) Add the following script to the notification
      
      Message: `${mail_script:cxs_EmailSearchResults}`

**Search as different user**

When you define contextual search for tables, you can choose to display search results as they would appear to a different user who performs the search.

For example, an HR administrator may want to display contextual search results, for a user who made an HR request, to find out why these results did not answer the user's query.

---

**Note**: The results for the user performing the search are based on that specific user's security access. Hence, the search result view for that user may have fewer entries than what that user can actually view.

---

1. Navigate to **Contextual Search > Table Configuration**.
2. Open a table configuration record.
3. In the **Search as** section of the form, fill in the fields as appropriate.
When enabled, Search As displays the search results that appear for the user defined in the Search As field.

Results appear in a separate tab next to the search results the current user sees.

Use Condition and Scripts to restrict when the other user’s results appear. Results appear if both checks evaluate to true:

- Condition in the Condition field evaluates to true, or the condition is empty.
- The script in the Script field evaluates to true, or sets the variable “answer” to true, or is empty.

<table>
<thead>
<tr>
<th>Enable search as</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Search as field</td>
<td>*</td>
</tr>
<tr>
<td>Results message</td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td></td>
</tr>
<tr>
<td>Script</td>
<td></td>
</tr>
</tbody>
</table>
Table 928: Search as configuration fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable search as</td>
<td>Check box use search as functionality.</td>
</tr>
<tr>
<td>Search as field</td>
<td>The field on the table to identify the user to search as. You can only select User Reference fields.</td>
</tr>
<tr>
<td></td>
<td>For example, select Opened by to search as the user who opened the relevant record.</td>
</tr>
<tr>
<td>Condition</td>
<td>Specify the conditions based on which the search as results are displayed.</td>
</tr>
<tr>
<td>Script</td>
<td>A condition script, enabling administrators to implement more powerful conditions.</td>
</tr>
<tr>
<td></td>
<td>For example, to restrict access to users that are members of a group, use gs.getUser().isMemberOf(&quot;&lt;group name&quot;).</td>
</tr>
</tbody>
</table>

**Note:** Both Condition and Script must evaluate to true for the results to be displayed.

**Note:** An empty script evaluates to true.

When an eligible user opens the form for this table and enters search terms, two tabs are displayed: one with the search results the user sees (My results), and one with the search results the other defined user sees (<User name>'s results).

**Define contextual search for record producer**

You can define a record producer to trigger a search based on text entered in the Comments field, and display search results at the bottom of the record producer.

For example, you can add contextual search to the Short description variable in the Ask a Question record producer to provide potential answers for users creating an incident using that record producer.

The default Create New Incident record producer implements contextual search. If you have customized this record producer, you can configure contextual search to link to a specific field on your record producer.

**Note:** Only one variable within a record producer can use contextual search.

1. Navigate to **Contextual Search > Search Contexts**.
2. Select the search context.
3. In the **Record Producer Configurations** related list, click **New**.
4. Complete the form.

Table 929: Configuration fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record producer</td>
<td>Name of the record producer to add the search to.</td>
</tr>
<tr>
<td>Search context</td>
<td>Name of the search context to use.</td>
</tr>
<tr>
<td>Search variable</td>
<td>Variable within the selected record producer which uses the contextual search.</td>
</tr>
<tr>
<td>Results header text</td>
<td>Label for the search results area that appears on the record producer.</td>
</tr>
<tr>
<td>Result action label</td>
<td>Label of the action button that appears when the user opens a search result record. By default this is set to This Helped.</td>
</tr>
<tr>
<td>Result action value</td>
<td>Internal value of the action button that appears when the user opens a search result record. By default this is set to helped. This value is stored within the Relevant Document [cxs_relevant_doc] table.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
Active | Check box for activating this record producer configuration.
Limit | Maximum number of items to display in the results area. By default, this is set to 10. To return an unrestricted number, leave this field blank.
Results per page | Number of results that are shown per page in the results area.

5. Click **Submit**.

**Define contextual search for wizard**

You can define contextual search for a wizard.

For example, if you have defined a wizard for creating incidents, you can add contextual search results to it.

---

**Note:** Wizards must be activated before you can use this functionality.

---

**Associate contextual search to a wizard**

You can associate contextual search to a text variable in a wizard to create a new wizard configuration record.

1. In the Search Context form, open the Wizard Configurations related list and click **New**. Alternatively, navigate to **Contextual Search > Wizard Configuration** and click **New**.
2. Fill in the fields, selecting an existing text variable to associate with contextual search.
3. Right-click the form header and select **Save**.
4. Click the **Add to Wizard** related link to associate the contextual search to the wizard.

**Add contextual search to wizard**

Update the wizard record to add the contextual search variable and implement contextual search.

1. Navigate to **System Wizards > Wizards**.
2. Open the wizard to be updated.
3. Define a new variable with **Type** set to **Macro** and **Macro** set to **cxs_wizard_search**.
4. Click **Submit**.

5. In the **Wizard Panels** related list, open the panel where you will add the macro variable.

   **Note:** Some types of panel may not allow variables.

6. On the **Variables** related list in the selected panel, click **Edit**.

7. Move the newly created macro from the **Collection** list to the **Variables List**.
8. Click **Save**.

By default, the Create Incident wizard includes contextual search linked to the **Please describe your symptoms** text variable.

### Apply configuration to search

You can apply configurations to define conditions for each search context. For example, you can configure the **catalog** search resource to only display search results from a specific service catalog.

1. In the **Search Context** form, open the **Resource Configuration** related list to show the resource configurations available for that context. The resource configurations available are based on the searcher selected for this search context.

2. Click the arrow icon beside the searcher name to expand the entry, showing which configuration properties are available.
### Search Context - Incident Deflection

<table>
<thead>
<tr>
<th>Record Producer Configurations (1)</th>
<th>Table Configurations (1)</th>
<th>Wizard Configurations (1)</th>
<th>Resource Configuration (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Search context configuration = Incident Deflection

<table>
<thead>
<tr>
<th>Go to</th>
<th>Name</th>
<th>Search resource configuration</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>catalog</td>
<td></td>
<td>true</td>
</tr>
</tbody>
</table>

#### Resource context config = catalog

<table>
<thead>
<tr>
<th>Go to</th>
<th>Name</th>
<th>Description</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Catalog Name</td>
<td>Restrict the search to only returning ...</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td>Search Operator</td>
<td>Overrides the default Catalog search res...</td>
<td>true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Go to</th>
<th>Name</th>
<th>Description</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>knowledge</td>
<td></td>
<td>true</td>
</tr>
</tbody>
</table>
3. Click a property name to view the property details.

Table 930: Search Properties

<table>
<thead>
<tr>
<th>Catalog Search Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog name</td>
<td>Restrict the search results displayed to those within the named catalog.</td>
</tr>
<tr>
<td>Search operator</td>
<td>Alter how to combine AND and OR conditions when parsing the search string, to score the results displayed. Set this to:</td>
</tr>
<tr>
<td></td>
<td>- IR_AND_OR_QUERY (the default): First display results with exact matches of all terms, then display results with any matches of any terms.</td>
</tr>
<tr>
<td></td>
<td>- IR_AND_QUERY: Display results with exact matches of all terms only.</td>
</tr>
<tr>
<td></td>
<td>- IR_OR_QUERY: Display results with any matches of any terms.</td>
</tr>
</tbody>
</table>
Tab 931: Name of form

<table>
<thead>
<tr>
<th>Knowledge Search Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge</td>
<td>enter an <em>encoded query string</em> defining the configuration required.</td>
</tr>
</tbody>
</table>

4. Click **Update**.

**Contextual search roles**

Contextual search involves the following roles.

<table>
<thead>
<tr>
<th>Role title [Name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator [admin]</td>
<td>Access and use contextual search functions.</td>
</tr>
</tbody>
</table>

**Deflect and resolve incidents with knowledge**

Incident deflection and resolution with knowledge provides contextual search results.

Contextual search is provided for the following areas:

- Incident record producers: deflecting incidents by helping end users resolve issues before they raise an incident.
- Incident forms: helping service desk staff resolve incidents quickly by providing relevant knowledge.
- Incident email notifications: helping end users resolve their incidents themselves without requiring manual intervention from service desk staff.

You can run feedback reports to track where these search results are marked as helpful.

You must have the admin role to configure contextual search.

**Provide knowledge in an incident form**

The Incident form displays contextual search results based on text entered in the **Short Description** field. These search results provide targeted knowledge to the incident analyst.

If the user who raised the incident has indicated that a returned result was useful, the incident analyst can review this information in more detail to help resolve the incident more quickly.
### Incident - INC0000055

<table>
<thead>
<tr>
<th>Number</th>
<th>SAP Sales and Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caller</td>
<td>Carol Coughlin</td>
</tr>
<tr>
<td>Location</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>-- None --</td>
</tr>
<tr>
<td>Subcategory</td>
<td>-- None --</td>
</tr>
<tr>
<td>Configuration Item</td>
<td>SAP Sales and Distribution</td>
</tr>
<tr>
<td>Impact</td>
<td>1 - High</td>
</tr>
<tr>
<td>Urgency</td>
<td>1 - High</td>
</tr>
<tr>
<td>Priority</td>
<td>1 - Critical</td>
</tr>
<tr>
<td>Assigned to</td>
<td>Beth Anglin</td>
</tr>
<tr>
<td>Opened</td>
<td>2014-10-16 21:47:23</td>
</tr>
<tr>
<td>Opened by</td>
<td>ITIL User</td>
</tr>
<tr>
<td>Contact type</td>
<td>Phone</td>
</tr>
<tr>
<td>State</td>
<td>Active</td>
</tr>
<tr>
<td>Assignment group</td>
<td>Service Desk</td>
</tr>
</tbody>
</table>

**Short description:** SAP Sales app is not accessible

**Related Search Results:**
- Can't access SAP
- SAP Outage - We are aware of the SAP outage, Service will be restored soon
- How to access office email from a home network
- Sales Force Automation is DOWN
- Permissions for Calendar, Email and Task

---

© 2017 ServiceNow. All rights reserved. 3027
You can **configure contextual search functionality** to change which fields are searched on or to add contextual search to other forms.

### Provide knowledge in incident record producer

Incident deflection provides self-service users with contextual knowledge when they create an incident using either the Create Incident record producer or the Something Broken ESS record producer.

Contextual search results appear based on text entered in the **Please describe your issue below** field of the Create Incident record producer, or in the Short description field of the Something Broken ESS record producer.

Users can indicate whether a search result is useful to them. If the knowledge provided resolved the incident, the user can cancel the record producer, which deflects the incident. The following scenario shows an example of this.

1. Navigate to the service catalog and select the **Can we help you?** category.
2. Select the **Create New Incident** catalog item to raise an incident.
   - The Create Incident record producer is displayed.

3. Contextual search then searches the text entered in the **Please describe your issue below** field to display knowledge results.
4. The user can click an entry in the results list to view the article.
5. The user can then indicate that the article helped, navigate to another article, or close the article.

6. If the knowledge provided helps resolve the issue, the user can cancel the incident record producer. The incident has been deflected. If the information did not fully resolve the issue, the user can still submit the incident.

You can configure contextual search functionality to adapt it to your organization’s incident record producers, or to add contextual search to other record producers. For example, customized incident record producers do not have contextual search enabled by default, so you can add contextual search to a customized record producer.

Provide knowledge in incident email notification

Contextual search results are included in email notifications that are sent to users who create a new incident.

This provides links to knowledge articles that may help users resolve their issues faster. For example, if a user raises an incident when the service desk staff is not available, the email notification provides knowledge links that may help the user.

By default, contextual search results are based on the short description in the incident. Within the automated email response, contextual search adds links to relevant knowledge articles.
For example:

1. Navigate to A customer sends an email to IT support with the subject *My laptop keeps crashing.*

2. An incident is created based on this email.

3. The email subject is inserted into the Short description field of the new incident.

4. The automated email notification sent to the user includes search results based on *My laptop keeps crashing.* For example, one article could be *How to upgrade your Windows version to stop your laptop crashing.*

5. The customer may then be able to use the returned results to resolve the incident.

---

**Note:**

The knowledge links provided are filtered to ensure the articles are accessible to the user who submitted the email.

---

By default, notifications provide three article links. You can configure notification options, such as changing the number of links provided with notifications, can be configured by administrators.

You can also configure contextual search functionality to match your organization’s email notifications or to use contextual search with notifications for other records.

**Attach an article**

If contextual search is available in a form that uses the knowledge search icon, that icon is removed to ensure there is only a single knowledge search mechanism for that form.

To retain the ability to attach a selected article to that form, an extra **Attach** button appears on the search results list, and an **Attach to incident** button appears on the displayed knowledge article.
Figure 800: Attach Button

Click the relevant button to attach that knowledge article to that incident, adding details of the knowledge article to the task.

Administer contextual search

You can configure contextual search functions. These procedures require the admin role.

Configure a property

Navigate to Contextual Search > Properties to view and edit contextual search properties.
### Table 933: Contextual search properties

<table>
<thead>
<tr>
<th>Property label [Name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value for maximum number of search results returned for table and record producer configurations [com.snc.contextual_search.result.default.limit]</td>
<td>Set an upper limit on the number of search results displayed in record producers or forms using contextual search. By default, this is set to 10.</td>
</tr>
<tr>
<td>Number of milliseconds that a configured field will wait before triggering a search [com.snc.search.service.wait_time]</td>
<td>Set the amount of time after a user finishes typing before running a search based on the typed text. By default, this is set to 500 (half a second).</td>
</tr>
<tr>
<td>Records returned above this threshold will log warnings [com.snc.contextual_search.result.threshold]</td>
<td>Trigger a warning message within the system log whenever more than this number of results is returned in a search. This logging helps if you are experiencing performance issues because of large searches. By default, this is set to 10,000.</td>
</tr>
<tr>
<td>When a Form configuration is created a search field with a name matching this property will be automatically created [com.snc.contextual_search.widget.form.default_field]</td>
<td>Link the context search to the search field identified in this property whenever you create a new context search for a form. This eliminates the need to select the field in the form manually. By default, this is set to short_description, meaning that whenever you create a new context search for a form with a short_description field, that field is automatically selected as the field the context search acts on.</td>
</tr>
<tr>
<td>Collapse the search results widget when opening an existing record in a form [com.snc.contextual_search.widget.form.open_collapsed_existing_records]</td>
<td>When set to true, the search results widget will be collapsed when an existing record is opened.</td>
</tr>
</tbody>
</table>

**Note:** This applies only to existing records and not new records.

---

### Run a feedback report

A **This Helped** button appears in displayed knowledge articles, allowing users to mark an article as helpful for resolving the associated issue. You can report on searches where a knowledge article was marked as helpful to measure the effectiveness of the contextual search results.

For example, service desk managers can produce a report on user issues that were resolved without an incident being raised to help measure the benefits of incident deflection using knowledge. This information can help to create targeted knowledge based on trends.

For example, analyzing commonly-raised incidents then creating knowledge articles based around those incident can help resolve or deflect similar incidents in future.

1. Navigate to **Reports > View / Run**.
2. Select the report to run.

Reports available by default are:

- **Number of deflected incidents over the last 3 months**: A line chart displaying the total number of deflected incidents per month for the last 3 months.
• **Successful deflection articles by search term**: A grouped list of search terms and the knowledge articles that were marked as helpful.

• **Top 15 most helpful articles**: A bar chart showing the top 15 articles that have been marked as helpful.

---

**Note**: You can also run a report by navigating to and searching for the word **Relevant**.

---

You can create additional feedback information reports by querying feedback tables.

**View a searcher**

Searchers specify where to locate information for a contextual search, such as knowledge and catalog items.

Searcher records are read-only, but you should be aware of the searchers available so you can select the correct one when defining contextual searches.

1. Navigate to **Contextual Search > Searchers**, then open a record.
2. Use the Search Resources related list to inspect the sources which define the information areas to search.
   For example, incident deflection uses the **knowledge and catalog** searcher which includes knowledge and catalog resources.

   Search resources can also contain properties, refining the resource further. For example, the knowledge search resource contains a Sort order property to specify that the search results are returned sorted by relevance.
Query feedback information

The information captured when the user clicks the **This Helped** action button on the result window is stored in the following tables.

<table>
<thead>
<tr>
<th>Table [name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant Document</td>
<td>Contains incident record references where users have indicated that a search result record was useful by clicking the <strong>This helped</strong> button</td>
</tr>
<tr>
<td>[cxs_relevant_doc]</td>
<td></td>
</tr>
<tr>
<td>Relevant Document Detail</td>
<td>Contains details of the search result record that users have indicated as useful or have attached to a form, as well as the search terms used to query the knowledge base.</td>
</tr>
<tr>
<td>[cxs_rel_doc_detail]</td>
<td></td>
</tr>
</tbody>
</table>

This feedback information is important for analyzing the effectiveness of the searches provided. You can create custom reports on this information by querying these tables.

Installed with contextual search

Activating the Contextual Search plugin installs the following components:

Business rules installed with contextual search

Contextual search adds the following business rules.

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add variable to record producer</td>
<td>Record Producer Configuration [cxs_rp_config]</td>
<td>Contains script that adds contextual search to the selected variable on a record producer.</td>
</tr>
<tr>
<td>Add variable to wizard</td>
<td>Wizard Configuration [cxs_wizard_config]</td>
<td>Contains script that adds contextual search to the selected variable on a wizard.</td>
</tr>
<tr>
<td>Cascading delete for removed config</td>
<td>Table Configuration [cxs_table_config]</td>
<td>Deletes search fields when the related table configuration is deleted.</td>
</tr>
<tr>
<td>Check for duplicate</td>
<td>Table Configuration [cxs_table_config]</td>
<td>Avoids duplicate table configurations.</td>
</tr>
<tr>
<td>Check for duplicate</td>
<td>Email Configuration [cxs_table_email_config]</td>
<td>Avoids duplicate table email configurations.</td>
</tr>
<tr>
<td>Check for duplicate</td>
<td>Wizard Configuration [cxs_wizard_config]</td>
<td>Avoids duplicate wizard configurations.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Check for duplicate</td>
<td>Record Producer Configuration [cxs_rp_config]</td>
<td>Avoids duplicate record producer configurations.</td>
</tr>
<tr>
<td>Check for invalid action values</td>
<td>Base UI Configuration [cxs_ui_config_base]</td>
<td>Checks for invalid action values against the related search configuration.</td>
</tr>
<tr>
<td>Create default search field</td>
<td>Table Configuration [cxs_table_config]</td>
<td>Creates a default search field when an insert or update is performed on a table configuration.</td>
</tr>
<tr>
<td>Create m2m_kb_task record on attach</td>
<td>Relevant Document Detail [cxs_rel_doc_detail]</td>
<td>Creates a knowledge base task record.</td>
</tr>
<tr>
<td>Delete m2m_kb_task record on remove</td>
<td>Relevant Document Detail [cxs_rel_doc_detail]</td>
<td>Deletes a knowledge base task record.</td>
</tr>
<tr>
<td>Display default context config message</td>
<td>Client Configuration [cxs_client_config]</td>
<td>Confirms the default search context configuration.</td>
</tr>
<tr>
<td>Display default field message</td>
<td>Search Field [cxs_table_field_config]</td>
<td>Confirms the default table field.</td>
</tr>
<tr>
<td>If searcher changes check interleaved</td>
<td>Search Context [cxs_context_config]</td>
<td>Check weather a search context needs to interleaved</td>
</tr>
<tr>
<td>Maintain context resource properties</td>
<td>Search Context [cxs_context_config]</td>
<td>Is used to update context properties</td>
</tr>
<tr>
<td>Make default</td>
<td>Search Context [cxs_context_config]</td>
<td>Sets the client configuration as the default.</td>
</tr>
<tr>
<td>Make default config</td>
<td>Search Field [cxs_table_field_config]</td>
<td>Sets the table field as the default.</td>
</tr>
<tr>
<td>Populate &quot;Order&quot; field</td>
<td>Search Field [cxs_table_field_config]</td>
<td>Sets the order for new search field configurations. Order will be set to the highest order in the list.</td>
</tr>
<tr>
<td>Remove related search field records</td>
<td>Table Configuration [cxs_table_config]</td>
<td>Removes related search fields when changes are made to a record producer</td>
</tr>
<tr>
<td>Remove related search field records</td>
<td>Record Producer Configuration [cxs_rp_config]</td>
<td>Removes related search fields when changes are made to a table.</td>
</tr>
<tr>
<td>Set &quot;Name&quot; field</td>
<td>Record Producer Configuration [cxs_rp_config]</td>
<td>Sets the name of a record automatically for record producers.</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Set &quot;Name&quot; field</td>
<td>Table Configuration</td>
<td>Sets the name of a record automatically for forms.</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_config]</td>
<td></td>
</tr>
<tr>
<td>Set &quot;Name&quot; field</td>
<td>Email Configuration</td>
<td>Sets the name of a record automatically.</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_email_config]</td>
<td></td>
</tr>
<tr>
<td>Set &quot;Name&quot; field</td>
<td>Search Field</td>
<td>Sets the name of a record automatically.</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_field_config]</td>
<td></td>
</tr>
<tr>
<td>Update related context config</td>
<td>Search Resource</td>
<td>Used to update related search context configuration records</td>
</tr>
<tr>
<td>records</td>
<td>[cxs_search_res_config]</td>
<td></td>
</tr>
<tr>
<td>Validate active context</td>
<td>Base UI Configuration</td>
<td>Used to confirm that the search context is valid</td>
</tr>
<tr>
<td></td>
<td>[cxs_ui_config_base]</td>
<td></td>
</tr>
<tr>
<td>Validate limit</td>
<td>Email Configuration</td>
<td>Ensures the limit is not negative or zero for email configurations.</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_email_config]</td>
<td></td>
</tr>
<tr>
<td>Validate limit</td>
<td>Record Producer Configuration</td>
<td>Ensures the limit is not negative or zero for record producer configurations.</td>
</tr>
<tr>
<td></td>
<td>[cxs_rp_config]</td>
<td></td>
</tr>
<tr>
<td>Validate limit</td>
<td>Table Configuration</td>
<td>Ensures the limit is not negative or zero for form configurations.</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_config]</td>
<td></td>
</tr>
<tr>
<td>Validate results per page</td>
<td>Table Configuration</td>
<td>Ensures the limit is not negative, zero or greater than limit for forms.</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_config]</td>
<td></td>
</tr>
<tr>
<td>Validate results per page</td>
<td>Record Producer Configuration</td>
<td>Ensures the limit is not negative, zero or greater than limit for record producers.</td>
</tr>
<tr>
<td></td>
<td>[cxs_rp_config]</td>
<td></td>
</tr>
</tbody>
</table>

Client scripts installed with contextual search

Contextual search adds the following client scripts.

Table 935: Client scripts installed with contextual search

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set default limit and results per page</td>
<td>Record Producer Configuration</td>
</tr>
<tr>
<td></td>
<td>[cxs_rp_config]</td>
</tr>
<tr>
<td>Clear field when Record Producer changes</td>
<td>Record Producer Configuration</td>
</tr>
<tr>
<td></td>
<td>[cxs_rp_config]</td>
</tr>
<tr>
<td>Set default limit and results per page</td>
<td>Table Configuration</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_config]</td>
</tr>
<tr>
<td>Name</td>
<td>Table</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Warn about search fields being removed</td>
<td>Table Configuration</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_config]</td>
</tr>
<tr>
<td>Set default limit</td>
<td>Email Configuration</td>
</tr>
<tr>
<td></td>
<td>[cxs_table_email_config]</td>
</tr>
<tr>
<td>Set default limit and results per page</td>
<td>Wizard Configuration</td>
</tr>
<tr>
<td></td>
<td>[cxs_wizard_config]</td>
</tr>
<tr>
<td>Clear field when Wizard changes</td>
<td>Wizard Configuration</td>
</tr>
<tr>
<td></td>
<td>[cxs_wizard_config]</td>
</tr>
</tbody>
</table>

Email notifications installed with contextual search

Contextual search modifies the following email notifications.

Table 936: Email notifications installed with contextual search

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident opened for me</td>
<td>Incident</td>
<td>Shows the search result records that contextual search finds for that incident record.</td>
</tr>
<tr>
<td></td>
<td>[Incident]</td>
<td></td>
</tr>
</tbody>
</table>

Note: A note indicates neutral or positive information that emphasizes or supplements important points of the main text. A note supplies information that may apply only in special cases. Examples are memory limitations, equipment configurations, or details that apply to specific versions of a program.

Script includes installed with contextual search

Table 937: Script includes installed with contextual search

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Extends</th>
</tr>
</thead>
<tbody>
<tr>
<td>cxs_App</td>
<td>Configures business rules, action scripts, and security script against tables using contextual search.</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_ContextConfig</td>
<td>Configures Table Configuration [cxs_table_config] attributes.</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_FormatResults</td>
<td>Determines and runs the macro to be used for displaying search results.</td>
<td>N/A</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Extends</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>cxs_Knowledge</td>
<td>Attaches knowledge articles to records such as incidents, change requests, and problems.</td>
<td>AbstractAjaxProcessor</td>
</tr>
<tr>
<td>cxs_RPAjax</td>
<td>Gets and retrieves record producer configurations.</td>
<td>AbstractAjaxProcessor</td>
</tr>
<tr>
<td>cxs_RPConfig</td>
<td>Configures the Record Producer Configuration [cxs_rp_config] table and manipulates contextual search on record producers.</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_RPVarTypes</td>
<td>Returns the types of variable that can be used when defining a contextual search for a record producer.</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_SearchResourceConfig</td>
<td>Contains method that will update Context Configuration for [cxs_search_res_config].</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_SearchResultsAJAX</td>
<td>Returns results from a search formatted using the specified macro.</td>
<td>AbstractAjaxProcessor</td>
</tr>
<tr>
<td>cxs_SearchServerAJAX</td>
<td>Checks that the search context is configured.</td>
<td>AbstractAjaxProcessor</td>
</tr>
<tr>
<td>cxs_TableActions</td>
<td>Gets email search request records.</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_TableAjax</td>
<td>Gets table configuration from the Table Configuration [cxs_table_config] table.</td>
<td>AbstractAjaxProcessor</td>
</tr>
<tr>
<td>cxs_TableConfig</td>
<td>Creates and configures fields that have contextual search attached from the cxs_table_field_config table.</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_TableEmailConfig</td>
<td>Configures and checks duplicates in the Email Configuration [cxs_table_email_config] table.</td>
<td>N/A</td>
</tr>
<tr>
<td>cxs_TableEmailConfigAJAX</td>
<td>Configures the Email Configuration [cxs_table_email_config] table.</td>
<td>AbstractAjaxProcessor</td>
</tr>
<tr>
<td>cxs_TableFieldConfig</td>
<td>Configures records in the Search Field [cxs_table_field_config] table.</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Tables installed with contextual search

Contextual search adds or modifies the following tables.

**Table 938: Tables installed with contextual search**

<table>
<thead>
<tr>
<th>Display Name [Table Name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Search Resource [v_cxs_search_resource]</td>
<td>A virtual table that contains a list of available search resources used in configuration of searches.</td>
</tr>
<tr>
<td>Email Configuration [cxs_table_email_config]</td>
<td>Contains a list of email notifications configured to have contextual search results attached.</td>
</tr>
<tr>
<td>Record Producer Configuration [cxs_rp_config]</td>
<td>Contains a list of record producers configured to have contextual search.</td>
</tr>
</tbody>
</table>
| Relevant Document [cxs_relevant_doc] | Contains incident record references where users have indicated that a search result record was useful by clicking the **This helped** button. Significant fields include:  
  - **Displayed on**: Whether the contextual search results were shown on a table form or a record producer.  
  - **Session id**: The id of the search session that the action button was triggered for.  
  - **Helped with**: The record that the search result helped resolve. |
<table>
<thead>
<tr>
<th>Display Name [Table Name]</th>
<th>Description</th>
</tr>
</thead>
</table>
| Relevant Document Detail [cxs_rel_doc_detail] | Contains all the search result records which users have indicated were useful or have attached to a form, as well as the search terms used in the search query. Significant fields include:  
  - **Relevance**: The internal value of the action button (This helped or Attach).  
  - **Help document**: The search result that was marked as helped or attached.  
  - **Help document URL**: The URL to the article that was marked as helped or attached.  
  - **Search request**: The JSON object that was used by the search engine to perform the search.  
  - **Search term**: The search term that was used to initiate the search. |
| Search Context [cxs_context_config]       | Contains available search contexts for contextual search.                                                                                   |
| Search Field [cxs_table_field_config]     | Contains all form fields which have contextual search configured to it.                                                                       |
| Search Resource Context Configuration [cxs_res_context_config] | Contains conditions for this search context that will restrict the results returned to a user.                                               |
| Search Resource Context Property [cxs_res_context_config_prop] | Contains the property values for search resource context configuration.                                                                     |
| Search Resource Property [cxs_search_res_config_prop] | Contains the property values for search resource searcher configuration.                                                                     |
| Searcher Configuration [cxs_searcher_config] | Contains the details of available search configurations.                                                                                     |
| Table Configuration [cxs_table_config]     | Lists the tables configured for table configurations.                                                                                         |
| Wizard Configuration [cxs_wizard_config]   | Lists the tables configured for wizard configurations.                                                                                         |
| Base UI Configuration [cxs_ui_config_base] | Contains the common elements for the record producer, table, and wizard configurations.                                                      |
UI macros installed with contextual search

Contextual search adds the following UI macros.

**Table 939: UI macros installed with contextual search**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cxs_result</td>
<td>Represents a single search result when processed by cxs_results.</td>
</tr>
<tr>
<td>cxs_results</td>
<td>Processes the results returned by a contextual search.</td>
</tr>
<tr>
<td>cxs_results_header_email</td>
<td>Adds the header when sending contextual search results by email.</td>
</tr>
<tr>
<td>cxs_results_header_rp</td>
<td>Displays the header that is shown above the search results on record producers.</td>
</tr>
<tr>
<td>cxs_results_header_table</td>
<td>Displays the header that is shown above the search results on forms.</td>
</tr>
<tr>
<td>cxs_results_header_wizard</td>
<td>Displays the header that is shown above the search results wizards.</td>
</tr>
<tr>
<td>cxs_results_table</td>
<td>Processes the results returned by a contextual search.</td>
</tr>
<tr>
<td>cxs_results_vcr</td>
<td>Provides the pagination controls.</td>
</tr>
<tr>
<td>cxs_result_default</td>
<td>Represents a single search result where there is no specific macro for the source table.</td>
</tr>
<tr>
<td>cxs_result_email</td>
<td>Represents a single search result when the search results are being sent by email.</td>
</tr>
<tr>
<td>cxs_result_kb_knowledge</td>
<td>Represents a single search result returned from the kb_knowledge table.</td>
</tr>
<tr>
<td>cxs_result_sc_cat_item</td>
<td>Represents a single search result returned from the sc_cat_item table.</td>
</tr>
<tr>
<td>cxs_result_table</td>
<td>Represents a single search result for displaying on a form.</td>
</tr>
<tr>
<td>cxs_rp_search</td>
<td>Initiates a contextual search from a record producer.</td>
</tr>
<tr>
<td>cxs_table_search</td>
<td>Initiates a contextual search from a form.</td>
</tr>
<tr>
<td>cxs_wizard_search</td>
<td>Initiates a contextual search from a record producer.</td>
</tr>
</tbody>
</table>

UI policies installed with contextual search

Contextual search adds the following UI policies.
Table 940: UI Policies installed with contextual search

<table>
<thead>
<tr>
<th>UI Policy</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide Table field</td>
<td>Email Configuration [cxs_table_email_config]</td>
<td>Changes email configuration visibility to false and read only.</td>
</tr>
<tr>
<td>Hide table config field</td>
<td>Search Field [cxs_table_field_config]</td>
<td>Hides the table field on the form.</td>
</tr>
</tbody>
</table>

Web services

HTTP-based web services allow diverse applications to talk to each other. ServiceNow supports both inbound (provider) and outbound (consumer) web services.

**Direct inbound web services**

Inbound web services allow you to access and modify ServiceNow data using a client application.

- REST API on page 3045
- Scripted REST APIs on page 3222
- SOAP web service on page 3255
- CSV Web service on page 3380
- EXCEL web service on page 3379
- JSONv2 Web Service on page 3380
- PDF web service on page 3391
- RSS feed generator on page 3395
- XML web service on page 3391

**Other inbound web services**

- ODBC driver on page 3470
- Scripted SOAP web services on page 3345

**Outbound web services**

Outbound web services allow you to send SOAP and REST messages to external web service providers.

- Outbound SOAP web service on page 3434
- Outbound REST web service on page 3403

**Inbound web services**

Inbound web services, such as the REST API, allow you to interact with ServiceNow instance data using web service requests.

ServiceNow supports inbound web services using REST and SOAP.
Additional web services are also available that provide specific functionality, such as the PDF or RSS web services.

REST API

REST (REpresentational State Transfer) is a simple stateless architecture that generally runs over HTTP. The REST style emphasizes that interactions between clients and services are enhanced by having a limited number of operations. Flexibility is provided by assigning resources their own unique universal resource indicators (URIs). Because each operation (GET, POST, PUT and DELETE) has a specific meaning, REST avoids ambiguity.

The REST API is active by default in all instances.

RESTful web services offer administrators several advantages, including:

- Support for different HTTP methods to perform different actions
- Detailed response codes and header information
- Pagination support for large data sets
- Streaming data on GET requests

Getting started with the REST API

To become familiar with ServiceNow REST API support, review these example scenarios and test the REST API functionality.

In this tutorial you will use the REST API Explorer to test the REST API. The REST API Explorer allows you to discover ServiceNow REST APIs, quickly construct and execute requests, and view responses from ServiceNow REST APIs within your browser.

Before beginning, ensure your user account has the rest_api_explorer and web_service_admin roles. These roles are required to complete the example procedures.

Getting started with REST - access the REST API Explorer

View available REST API resources using the REST API Explorer.

Role required: web_service_admin, rest_api_explorer, or admin

You can browse available APIs, API versions, and methods for each API.

Navigate to System Web Services > REST API Explorer.
### Table API

Allows you to perform create, read, update and delete (CRUD) operations on existing tables

**Retrieve records from a table**

GET `https://<hostname>.service-now.com/api/now/table/tableName`

<table>
<thead>
<tr>
<th><strong>Prepare request</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Request Headers</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Request Format</td>
</tr>
<tr>
<td>Response Format</td>
</tr>
<tr>
<td>Authorization</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Path Parameters</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>tableName</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Query Parameters</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>sysparm_query</td>
</tr>
<tr>
<td>sysparm_display_value</td>
</tr>
<tr>
<td>sysparm_exclude_reference_link</td>
</tr>
</tbody>
</table>
Getting started with REST - retrieve existing incidents

Use a GET request to view existing incident records.

Role required: web_service_admin, rest_api_explorer, or admin

Use the REST API Explorer to send the following request:

GET https://instance.service-now.com/api/now/v1/table/incident

1. In the top-left of the REST API Explorer, select **Table API** and **v1** version.
2. Click **Retrieve records from a table (GET)**.
3. In the **Path Parameters** section, select the Incident table.
4. Scroll to the bottom of the page and click **Send**.

The response includes incident records from the instance. The REST API Explorer limits queries to 10 records at a time, only the first 10 incident records appear. The response also includes a Link header that provides the URL to query the next 10 incident records.
Response

Status code: 200 OK

Headers
- Content-Encoding: gzip
- Content-Type: application/json
- Date: Tue, 08 Oct 2015 08:24:46 GMT
- Server: ServiceNow
- Transfer-Encoding: chunked
- X-Total-Count: 54

Response Body
{
  "result": [
    {
      "location": {
        "Link": "https://service-now.com/api/now/table/cmn_location/1083351cc611227501b682158cabf646",
        "value": "1083351cc611227501b682158cabf646"
      },
      "expected_start": "",
      "reopen_count": "",
      "close_notes": "",
      "impact": "1",
      "urgency": "1",
      "correlation_id": "",
      "sys_tags": "",
      "sys_domain": {
        "Link": "https://service-now.com/api/now/table/sys_user_group/global",
      }
    }
  ]
}
Getting started with REST - create an incident record
Use a POST request to create a new record.

Role required: web_service_admin, rest_api_explorer, or admin

Use the REST API Explorer to send the following request:

POST https://instance.service-now.com/api/now/v1/table/incident

1. In the top-left of the REST API Explorer, click Create a record (POST).
2. In the Path Parameters section, select the Incident table.
3. In the Request Body section, click Add a field.
4. Select a field and specify a value for that field.
5. Optional: Click Add another field and specify an additional field to assign a value to.
   The request body updates automatically based on your entries, such as
   {"short_description":"Test incident creation through REST",
   "comments":"These are my comments"}
6. After constructing the request, click Send.
   The response includes a Location header that specifies where the incident was created and how to
   retrieve the incident. Record this header to use in the next part of this guide.
**Response**

- **Status code**: 201 Created

**Headers**
- Content-Encoding: gzip
- Content-Type: application/json
- Date: Tue, 08 Sep 2015 12:11:03 GMT
- Location: https://[redacted].servicenow.com/api/now/v1/table/incident/ba8f71b22b1a02006706b7219da154f
- Server: ServiceNow
- Transfer-Encoding: chunked

**Response Body**

```
"short_description": "Test incident creation through REST",
"order": "",
"sys_updated_by": "admin1",
"resolved_by": "",
"notify": "1",
"upon_reject": "cancel",
"approval_history": "",
"problem_id": "",
"work_notes": "",
"calendar_duration": "",
"close_code": "",
"sys_id": "ba8f71b22b1a02006706b7219da154f",
```

---

*Getting started with REST - read the inserted incident*

Use the Location header from the POST response to run a GET request.

Role required: web_service_admin, rest_api_explorer, or admin
Use the REST API Explorer to send the following request:

GET https://instance.service-now.com/api/now/v1/table/incident/(sys_id)

1. In the top-left of the REST API Explorer, click **Retrieve a record (GET)**.
2. In the **Path Parameters** section, select the Incident table.
3. In the **sys_id** field, enter the `sys_id` of the record you created.
   
   The `sys_id` appears as a 32-character string at the end of the POST response Location header.
4. Click **Send**.

   The response body contains a text representation of the record. You can control the format of the response, such as JSON or XML, using the **Response Format** field.
Getting started with REST - update the incident

Update the incident record using either a PUT or PATCH function.

Role required: web_service_admin, rest_api_explorer, or admin

Use the REST API Explorer to send the following request:
PUT https://instance.service-now.com/api/now/v1/table/incident/(sys_id)?sysparm_exclude_ref_link=true

1. In the top-left of the REST API Explorer, click Modify a record (PUT) or Update a record (PATCH).
2. In the Path Parameters section, select the Incident table.
3. In the sys_id field, enter the sys_id of the record you created.
4. In the Request Body section, click Add a field.
5. Select the Short description field and specify a new value.
6. Click Send.
7. Verify that the response contains the updated short_description value.
Getting started with REST - delete the incident

Delete the incident using a DELETE request.

Role required: web_service_admin, rest_api_explorer, or admin

Use the REST API Explorer to send the following request:

DELETE https://instance.service-now.com/api/now/v1/table/incident/(sys_id)

1. In the top-left of the REST API Explorer, click Delete a record (DELETE).
2. In the Path Parameters section, select the Incident table.
3. In the **sys_id** field, enter the sys_id of the record you created.
4. Click **Send**.
5. Verify that the response status code is 204.

![Response Table]

### Available REST APIs

The inbound REST API provides multiple endpoints and methods.

To view the the REST API resources available in your instance, along with the correct formats for the supported methods, navigate to **System Web Services > REST API Explorer**.

**REST API explorer**
The REST API Explorer uses information from your instance to provide you with a list of endpoints, methods, and variables that you can use to build and send a REST request.

After you build the request, the REST API Explorer provides code samples in multiple programming languages that you can use to send the request, and detailed request and response information.

Access the REST API Explorer by navigating to **System Web Services > REST API Explorer**. You must have the rest_api_explorer role to access the REST API Explorer.

**Warning:** The REST API Explorer interacts with data on the current instance. Use caution when working with requests that create, edit, or delete data on a production instance.

### Explore the REST API for a table

You can explore the REST API for a table directly from that table. Explore the API using the REST API Explorer to quickly construct and test REST requests for that table.
Role required: itil, personalize_dictionary, and rest_api_explorer

1. Navigate to any form or list.
2. Right-click the form or list header and select Configure > Table.
3. Click the Explore REST API related link.

The REST API Explorer opens, displaying the Table API for the selected table. If the table does not allow web service interaction, a warning appears instead.

Use the REST API Explorer to construct and test REST requests for the table.

REST API explorer elements
The REST API Explorer displays multiple elements that allow you to construct and test REST requests.
The Namespace, API, method, Request Headers and Path Parameters sections appear for all requests. Other sections may appear depending on the selected API and method.

Namespace, API, and method

The choice lists at the top of the REST API Explorer allow you to select an API, such as the Table API, Aggregate API, or Import Set API, and an API version. Available APIs depend on the selected Namespace. REST APIs provided by ServiceNow use the now namespace. Scripted REST Services may use a different namespace.

Links that appear below these choice lists display available methods for the selected API and version, such as the GET method for the Table API. Each method includes a brief description of the action that method performs. The REST API Explorer displays a sample endpoint and certain Request Elements based on the selected API and method.

Request headers

In the Request Headers section, specify the HTTP request and response headers and choose to send the request as yourself or as another user. If you select another user, enter the username and password.

To add a custom header, click Add custom header and specify the header name and value.

Path parameters

In the Path Parameters section, select the table to interact with. For the Import Set API, import staging tables are available. For the Table API and Aggregate API, all tables that allow web service interaction are available. For requests that interact with a single existing record, such as those that modify a record, the sys_id field allows you to specify the record.

Query parameters

In the Query Parameters section, define parameters to modify the request, such as the encoded query used to filter results when performing a Table API GET request.

The parameters that appear depend on the selected request method. To add custom query parameters, click Add query parameter and specify the parameter name and value. Refer to the documentation for each REST API method for detailed information about the available parameters.
Request body

In the Request Body section, define the body content of the request. This section appears for methods that insert or update data, such as PUT. The Request Format that you select in the Request Headers section determines the format you must use in the request body, such as JSON or XML.

You can build the request body by selecting fields, by manually writing the request body, or by uploading a binary file.

Use the Builder tab to build the request body using instance fields. Click Add a field, then select a field and specify a value. Available fields depend on the table you are adding data to.

Use the Raw tab to manually enter a request body.

Use the Binary tab to upload a binary file. This tab appears only for APIs that support binary file data, such as the Attachment API.

Code samples

You can obtain a code sample in various languages that you can use to send the request. Links for code samples appear at the bottom of the form. Click a link to view or copy the code sample for that language. The code samples change based on data you enter on the form. Code samples are available in these languages:

- ServiceNow Script
- cURL
- Python
- Ruby
- JavaScript
- Perl
- Powershell

Request and response details

After sending a request, the Request and Response sections appear providing detailed information including the full request URI, headers for both the request and response, the HTTP status code of the response, and the response body. Click Clear Response to hide this information.

When testing a request that returns a binary response, such as a file attachment, the REST API Explorer uses the browser default functionality to handle the file, such as by prompting you to download the file.

**Table API**
The Table API allows you to perform create, read, update, and delete (CRUD) operations on existing tables.

Version differences

The default version of the Table API is v2. You may notice the following differences between the current and previous versions.

- Version 1 of the Table API returns the error code 404 on a valid GET query that contains no results. Version 2 returns the success status code 200 and an empty array as the response body instead.

Table API - GET /now/table/{tableName}
This method retrieves multiple records for the specified table with proper pagination information.

**URL format**

Versioned URL: `/api/now/v2/table/{tableName}`
Default URL: `/api/now/table/{tableName}`

**Supported request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_query</td>
<td>Encoded query used to filter the result set. For example: <code>(sysparm_query=caller_id=javascript:gs.getUserID()^active=true)</code> The encoded query supports order by. To sort responses based on certain fields, use the ORDERBY and ORDERBYDESC clauses in sysparm_query. For example, <code>sysparm_query=active=true^ORDERBYnumber^ORDERBYDESCcategory</code> filters all active records and orders the results in ascending order by number first, and then in descending order by category. If part of the query is invalid, such as by specifying an invalid field name, the instance ignores the invalid part. It then returns rows using only the valid portion of the query. You can control this behavior using the property glide.invalid_query.returns_no_rows. Set this property to true to return no rows on an invalid query.</td>
</tr>
</tbody>
</table>

**Note:** This property controls the behavior of all queries across the instance, such as in lists, scripts (GlideRecord.query()), and web service APIs.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_display_value</td>
<td>Data retrieval operation for reference and choice fields.</td>
</tr>
<tr>
<td></td>
<td>Based on this value, retrieves the display value and/or the actual value from the database.</td>
</tr>
<tr>
<td></td>
<td>• true  returns display values for all fields.</td>
</tr>
<tr>
<td></td>
<td>• false  returns actual values from the database.</td>
</tr>
<tr>
<td></td>
<td>If a value is not specified, this parameter defaults to false.</td>
</tr>
<tr>
<td></td>
<td>• all  returns both actual and display values.</td>
</tr>
<tr>
<td>Note:</td>
<td>There is no preferred method for setting this parameter. However, specifying the display value may cause performance issues since it is not reading directly from the database and may include referencing other fields and records. For more information on display values and actual values, see Table API FAQs (KB0534905).</td>
</tr>
<tr>
<td>sysparm_fields</td>
<td>Comma-separated field names to return in the response.</td>
</tr>
<tr>
<td>sysparm_view</td>
<td>UI view to determine fields returned in the response.</td>
</tr>
<tr>
<td>Note:</td>
<td>If both sysparm_fields and sysparm_view are specified, the sysparm_fields parameter takes priority.</td>
</tr>
<tr>
<td>sysparm_limit</td>
<td>Limit to be applied on pagination. The default is 10000.</td>
</tr>
<tr>
<td></td>
<td>Unusually large sysparm_limit values can impact system performance.</td>
</tr>
<tr>
<td>sysparm_offset</td>
<td>Number of records to exclude from the query. Use this parameter to obtain more records than specified in sysparm_limit. For example, if sysparm_limit is set to 500, but there are additional records you want to query, specify a sysparm_offset value of 500 to get the second set of records. Do not pass a negative number in the sysparm_offset parameter.</td>
</tr>
<tr>
<td>sysparm_exclude_reference_link</td>
<td>Additional information provided for reference fields, such as the URI to the reference resource, is suppressed.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sysparm_read_replica_category</td>
<td>Category value to read data from read replicas.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This parameter is not commonly used. The instance must have read replica support.</td>
</tr>
<tr>
<td>sysparm_suppress_pagination_header</td>
<td>Set this value to true to remove the Link header from the response. The Link header enables you to request additional pages of data when the number of records matching your query exceeds the query limit.</td>
</tr>
</tbody>
</table>

**Key-Value Pairs**

Alternative to using the sysparm_query parameter. You can filter a query using key-value pairs where the key is the name of a field. This functionality is available for GET queries.

For example, instead of using the parameter `&sysparm_query=active=true`, you can use `&active=true`. You can use the display value when the field is a choice or reference type field, such as `&state=closed` instead of `&state=7`. To specify multiple key-value pairs, separate each with an ampersand, such as `&active=true&assigned_to=john.smith`.

---

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

**Table 942: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
### Table 943: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
</table>
| Link      | REST message data can be split into multiple result sets rather than forcing the user to submit multiple requests. The header has different links available for the first set, previous set, next set, and the last set of records, where applicable. For example:  

https://<instance name>.service-now.com/api/now/table/cmdb_ci?sysparm_offset=40&sysparm_limit=10000>;rel="next"  
https://<instance name>.service-now.com/api/now/table/cmdb_ci?sysparm_offset=40&sysparm_limit=10000>;rel="prev"  
https://<instance name>.service-now.com/api/now/table/cmdb_ci?sysparm_offset=0&sysparm_limit=10000>;rel="first"  
https://<instance name>.service-now.com/api/now/table/cmdb_ci?sysparm_offset=2780&sysparm_limit=10000>;rel="last"  

**Note:** The limit parameter defaults to 10,000 records. This limit can be set to any value. Be aware, however, that an unusually large value can impact system performance. |
| X-Total-Count | Total count of records returned by the query. |

### Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see [REST API response codes](#).

**Table 944: Status codes**

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Request completed successfully. If a valid query returned no results, the response body contains only an empty result array.</td>
</tr>
</tbody>
</table>

### Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/table/problem?
sysparm_limit=1" \
```
```json
{
    "result": [
        {
            "parent": "",
            "made_sla": "true",
            "watch_list": "",
            "upon_reject": "cancel",
            "sys_updated_on": "2016-01-19 04:52:04",
            "approval_history": "",
            "number": "PRB0000050",
            "sys_updated_by": "glide.maint",
            "opened_by": {
                "link": "https://instance.service-now.com/api/now/table/sys_user/gluck.maint",
                "value": "glide.maint"
            },
            "user_input": "",
            "sys_created_on": "2016-01-19 04:51:19",
            "sys_domain": {
                "link": "https://instance.service-now.com/api/now/table/sys_user_group/global",
                "value": "global"
            },
            "state": "4",
            "sys_created_by": "glide.maint",
            "knowledge": "false",
            "order": "",
            "closed_at": "2016-01-19 04:52:04",
            "cmdb_ci": {
                "link": "https://instance.service-now.com/api/now/table/cmdb_ci/55b35562c0a8010e01ccfe22378e0aea9",
                "value": "55b35562c0a8010e01ccfe22378e0aea9"
            },
            "delivery_plan": "",
            "impact": "3",
            "active": "false",
            "work_notes_list": "",
            "business_service": "",
            "priority": "4",
            "sys_domain_path": "/",
            "time_worked": "",
            "expected_start": "",
            "rejection_goto": "",
            "opened_at": "2016-01-19 04:49:47",
            "business_duration": "1970-01-01 00:00:00",
            "group_list": "",
            "work_end": "",
            "approval_set": "",
            "wf_activity": "",
            "work_notes": "",
            "short_description": "Switch occasionally drops connections",
            "correlation_display": "",
            "delivery_task": "",
            "work_start": "",
            "assignment_group": "",
            "additional_assignee_list": "",
            "description": "Switch occasionally drops connections",
            "calendar_duration": "1970-01-01 00:02:17"
        }
    ]
}
```
Sample Python request

```python
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/table/problem?sysparm_limit=1'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'username'
pwd = 'password'

# Set proper headers
headers = {'Content-Type': 'application/json', 'Accept': 'application/json'}

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:', response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>
    "close_notes": "updated firmware",
    "sys_class_name": "problem",
    "closed_by": "",
    "follow_up": "",
    "sys_id": "04ce72c9c0a8016600b57f75ac67b5b",
    "contact_type": "phone",
    "urgency": "3",
    "company": "",
    "reassignment_count": "",
    "activity_due": "",
    "assigned_to": "",
    "comments": "",
    "approval": "not requested",
    "sla_due": "",
    "comments_and_work_notes": "",
    "due_date": "",
    "sys_mod_count": "1",
    "sys_tags": "",
    "escalation": "0",
    "upon_approval": "proceed",
    "correlation_id": "",
    "location": ""
  }
}
```
Can't read email

Closed/Resolved by Caller

https://instance.service-now.com/api/now/table/sys_user_group/d625dcce0ca8016700a222a0f7900d06

https://instance.service-now.com/api/now/table/sys_user/9ee1b13dc6112271007f9d0efdb69cd0

https://instance.service-now.com/api/now/table/problem/9d3a266ac6112287004e37fb2ceb0133

https://instance.service-now.com/api/now/table/sys_user/46b87022a9fe198101a78787e40d7547

https://instance.service-now.com/api/now/table/cmn_location/1083361cc611227501b682158cabf646
Table API - POST /now/table/{tableName}
This method inserts one record in the specified table. Multiple record insertion is not supported by this method.

URL format

Versioned URL: /api/now/v2/table/{tableName}
Default URL: /api/now/table/{tableName}

Supported request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_display_value</td>
<td>Data retrieval operation for reference and choice fields.</td>
</tr>
<tr>
<td></td>
<td>Based on this value, retrieves the display value and/or the actual value from the database.</td>
</tr>
<tr>
<td></td>
<td>• true returns display values for all fields.</td>
</tr>
<tr>
<td></td>
<td>• false returns actual values from the database.</td>
</tr>
<tr>
<td></td>
<td>• If a value is not specified, this parameter defaults to false.</td>
</tr>
<tr>
<td></td>
<td>• all returns both actual and display values.</td>
</tr>
</tbody>
</table>

**Note:** There is no preferred method for setting this parameter. However, specifying the display value may cause performance issues since it is not reading directly from the database and may include referencing other fields and records. For more information on display values and actual values, see *Table API FAQs (KB0534905).*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_fields</td>
<td>Comma-separated field names to return in the response.</td>
</tr>
<tr>
<td>sysparm_view</td>
<td>UI view to determine fields returned in the response.</td>
</tr>
</tbody>
</table>

**Note:** If both sysparm_fields and sysparm_view are specified, the sysparm_fields parameter takes priority.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_exclude_reference_link</td>
<td>Additional information provided for reference fields, such as the URI to the reference resource, is suppressed.</td>
</tr>
<tr>
<td>sysparm_input_display_value</td>
<td>Data insert or update operations.</td>
</tr>
<tr>
<td></td>
<td>For values in the request:</td>
</tr>
<tr>
<td></td>
<td>• true treats input values as display values and they are manipulated so they can be stored properly in the database.</td>
</tr>
<tr>
<td></td>
<td>• false treats input values as actual values and stored them in the database without manipulation.</td>
</tr>
</tbody>
</table>

**Note:**
- If this parameter is set to true, pay attention to input values, especially date values, as these are interpreted as being supplied via the user time zone preference and are transformed into UTC format.
- To set the value of an encrypted field, you must set this parameter to true. If this parameter is not set to true, values submitted to encrypted fields are not saved. Additionally, the requesting user must have the appropriate encryption context prior to submitting the request. Encrypted fields are hidden for users without the appropriate encryption context. For more information on display values and actual values, see Table API FAQs (KB0534905). For more information on field encryption see Use an encrypted text field on page 2532.

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.
### Table 946: Request headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-no-response-body</td>
<td>By default, responses include body content detailing the new record. Set this header to true in the request to suppress the response body.</td>
</tr>
<tr>
<td>Content-Type</td>
<td>The data format of the request body.</td>
</tr>
</tbody>
</table>

### Table 947: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>The location of the created resource.</td>
</tr>
</tbody>
</table>

### Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see REST API response codes.

### Table 948: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Indicates the request completed successfully.</td>
</tr>
</tbody>
</table>

### Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/table/incident" \
--request POST \
--header "Accept:application/json"\n--header "Content-Type:application/json" \
--data "{'short_description':'Unable to connect to office wifi','assignment_group':'287ebd7da9fe198100f92cc8d1d2154e','urgency':'2','impact':'2'}" \
--user 'admin':'admin'
```
Unable to connect to office wifi

sys_user_group/:global
"sys_user_group/:global"
Sample Python request

```python
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/table/incident'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}

# Do the HTTP request
response = requests.post(url, auth=(user, pwd),
headers=headers, data="<request><entry><short_description>Unable to connect to office wifi</short_description><assignment_group>287ebd7da9fe198100f92cc8d1d2154e</assignment_group><urgency>2</urgency><impact>2</impact></entry></request>"

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:',response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>
    <upon_approval>proceed</upon_approval>
  </result>
</response>
```
Unable to connect to office wifi
<upon_reject>cancel</upon_reject>
<approval_history />
<problem_id />
<calendar_duration />
<work_notes />
<close_code />
<approval>not requested</approval>
<sys_id>d977b66a4f411200adf9f8e18110c7b2</sys_id>
<caused_by />
<severity>3</severity>
<sys_created_by>admin</sys_created_by>
<assigned_to />
<resolved_at />
<business_stc />
<cmdb_ci />
<sys_domain_path>/</sys_domain_path>
<wf_activity />
<opened_by>
  <link>https://instance.service-now.com/api/now/table/sys_user/6816f79cc0a8016401c5a33be04be441</link>
  <value>6816f79cc0a8016401c5a33be04be441</value>
</opened_by>
<rejection_goto />
<subcategory />
<sys_class_name>incident</sys_class_name>
<watch_list />
<escalation>0</escalation>
<contact_type>phone</contact_type>
<time_worked />
<comments />
</result>
</response>

Table API - GET /now/table/{tableName}/{sys_id}
This method retrieves the record identified by the specified sys_id from the specified table.

**URL format**

Versioned URL: /api/now/v2/table/{tableName}/{sys_id}
Default URL: /api/now/table/{tableName}/{sys_id}
Supported request parameters

**Table 949: Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_display_value</td>
<td>Data retrieval operation for reference and choice fields. Based on this value, retrieves the display value and/or the actual value from the database.</td>
</tr>
<tr>
<td></td>
<td>• true returns display values for all fields.</td>
</tr>
<tr>
<td></td>
<td>• false returns actual values from the database. If a value is not specified, this parameter defaults to false.</td>
</tr>
<tr>
<td></td>
<td>• all returns both actual and display values.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> There is no preferred method for setting this parameter. However, specifying the display value may cause performance issues since it is not reading directly from the database and may include referencing other fields and records. For more information on display values and actual values, see Table API FAQs (KB0534905).</td>
</tr>
<tr>
<td>sysparm_fields</td>
<td>Comma-separated field names to return in the response.</td>
</tr>
<tr>
<td>sysparm_view</td>
<td>UI view to determine fields returned in the response.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If both sysparm_fields and sysparm_view are specified, the sysparm_fields parameter takes priority.</td>
</tr>
<tr>
<td>sysparm_exclude_reference_link</td>
<td>Additional information provided for reference fields, such as the URI to the reference resource, is suppressed.</td>
</tr>
</tbody>
</table>

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

**Table 950: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
### Table 951: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

### Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see [REST API response codes](#).

#### Table 952: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the request completed successfully.</td>
</tr>
<tr>
<td>404</td>
<td>Indicates the record is not found or the requesting user does not have access to the record.</td>
</tr>
</tbody>
</table>

#### Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/table/incident/a9e30c7dc61122760116894de7bcc7bd" \
--request GET \
--header "Accept:application/json" \
--user 'username':'password'
```

```json
{
    "result": {
        "upon_approval": "",
        "location": {
            "link": "https://instance.service-now.com/api/now/table/cmn_location/105cf7f3c611227501e75e08b14a38ba",
            "value": "105cf7f3c611227501e75e08b14a38ba"
        },
        "expected_start": "",
        "reopen_count": "",
        "close_notes": "",
        "additional_assignee_list": "",
        "impact": "1",
        "urgency": "3",
        "correlation_id": "",
        "sys_tags": "",
        "sys_domain": {
            "link": "https://instance.service-now.com/api/now/table/sys_user_group/global",
            "value": "global"
        },
        "description": "",
        "group_list": "",
        "priority": "3",
        "delivery_plan": ""
    }
}
```
Sample Python request

```python
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/table/incident/a9e30c7dc61122760116894de7bcc7bd'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:',response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```
Supported request parameters

Table 953: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_display_value</td>
<td>Data retrieval operation for reference and choice fields. Based on this value, retrieves the display value and/or the actual value from the database. • true returns display values for all fields. • false returns actual values from the database. If a value is not specified, this parameter defaults to false. • all returns both actual and display values.</td>
</tr>
<tr>
<td>sysparm_fields</td>
<td>Comma-separated field names to return in the response.</td>
</tr>
<tr>
<td>sysparm_view</td>
<td>UI view to determine fields returned in the response.</td>
</tr>
<tr>
<td>sysparm_exclude_reference_link</td>
<td>Additional information provided for reference fields, such as the URI to the reference resource, is suppressed.</td>
</tr>
</tbody>
</table>

**Note:** There is no preferred method for setting this parameter. However, specifying the display value may cause performance issues since it is not reading directly from the database and may include referencing other fields and records. For more information on display values and actual values, see Table API FAQs (KB0534905).
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| sysparm_input_display_value | Data insert or update operations. For values in the request:  
• true treats input values as display values and they are manipulated so they can be stored properly in the database.  
• false treats input values as actual values and stored them in the database without manipulation.  

**Note:**  
• If this parameter is set to true, pay attention to input values, especially date values, as these are interpreted as being supplied via the user time zone preference and are transformed into UTC format.  
• To set the value of an encrypted field, you must set this parameter to true. If this parameter is not set to true, values submitted to encrypted fields are not saved. Additionally, the requesting user must have the appropriate encryption context prior to submitting the request. Encrypted fields are hidden for users without the appropriate encryption context. For more information on display values and actual values, see Table API FAQs (KB0534905). For more information on field encryption see Use an encrypted text field on page 2532. |

### Headers

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

**Table 954: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-no-response-body</td>
<td>By default, responses include body content detailing the modified record. Set this header to true in the request to suppress the response body.</td>
</tr>
</tbody>
</table>
Geneva  ServiceNow  ServiceNow Platform

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>The data format of the request body.</td>
</tr>
</tbody>
</table>

Table 955: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see REST API response codes.

Table 956: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the request completed successfully.</td>
</tr>
</tbody>
</table>

Sample cURL request

```
curl "https://instance.service-now.com/api/now/table/incident/ef43c6d40a0b5700c77f9bf387afe3" \
  --request PUT \ 
  --header "Accept:application/json"\ 
  --header "Content-Type:application/json" \ 
  --data "{"assigned_to':"681b365ec0a80164000fb0b05854a0cd','urgency':'1','comments':'Elevating urgency, this is a blocking issue'}" \ 
  --user 'admin':'admin'
```

```json
{
  "result": {
    "upon_approval": "proceed",
    "location": {
      "link": "https://instance.service-now.com/api/now/table/cmn_location/108752c8c611227501d4ab0e392ba97f",
      "value": "108752c8c611227501d4ab0e392ba97f"
    },
    "expected_start": "",
    "reopen_count": "",
    "close_notes": "",
    "additional_assignee_list": "",
    "impact": "1",
    "urgency": "1",
    "correlation_id": "",
    "sys_tags": "",
    "sys_domain": {
      "link": "https://instance.service-now.com/api/now/table/sys_user_group/global",
      "value": "global"
    }
  }
}
```
Can't access Exchange server - is it down?
"close_code": "",
"sys_id": "ef43c6d40a0a0b5700c77f9bf387afe3",
"approval": "not requested",
"caused_by": "",
"severity": "3",
"sys_created_by": "glide.maint",
"resolved_at": "",
"assigned_to": {
  "link": "https://instance.service-now.com/api/now/table/sys_user/681b365ec0a80164000fb0b05854a0cd",
  "value": "681b365ec0a80164000fb0b05854a0cd"
},
"business_stc": "",
"wf_activity": "",
"sys_domain_path": "/",
"cmdb_ci": {
  "link": "https://instance.service-now.com/api/now/table/cmdb_ci/281190e3c0a8000b003f593aa3f20ca6",
  "value": "281190e3c0a8000b003f593aa3f20ca6"
},
"opened_by": {
  "link": "https://instance.service-now.com/api/now/table/sys_user/glide.maint",
  "value": "glide.maint"
},
"subcategory": "",
"rejection_goto": "",
"sys_class_name": "incident",
"watch_list": "",
"time_worked": "",
"contact_type": "phone",
"escalation": "3",
"comments": ""
}

Sample Python request

# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/table/incident/ef43c6d40a0a0b5700c77f9bf387afe3'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}

# Do the HTTP request
response = requests.put(url, auth=(user, pwd), headers=headers, data="<request><entry><assigned_to>681b365ec0a80164000fb0b05854a0cd</assigned_to><urgency>1</urgency><comments>Elevating urgency, this is a blocking issue</comments></entry></request>")

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:',
    response.headers, 'Error Response:', response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)

<?xml version="1.0" encoding="UTF-8"?>
<response>
   <result>
      <upon_approval>proceed</upon_approval>
      <location>
         <link>https://instance.service-now.com/api/now/table/cmnication/108752c8c611227501d4ab0e392ba97f</link>
         <value>108752c8c611227501d4ab0e392ba97f</value>
      </location>
      <expected_start />
      <reopen_count />
      <close_notes />
      <additional_assignee_list />
      <impact>1</impact>
      <urgency>1</urgency>
      <correlation_id />
      <priority>1</priority>
      <sys_tags />
      <sys_domain>
         <link>https://instance.service-now.com/api/now/table/sys_user_group/global</link>
         <value>global</value>
      </sys_domain>
      <description />
      <group_list />
      <delivery_plan />
      <sys_mod_count>8</sys_mod_count>
      <work_notes_list />
      <follow_up />
      <business_service />
      <closed_at />
      <sla_due>2017-07-05 05:58:24</sla_due>
      <delivery_task />
      <sys_updated_on>2016-01-22 14:14:54</sys_updated_on>
      <parent />
      <work_end />
      <number>INC0000050</number>
      <closed_by />
      <work_start />
      <calendar_stc />
      <category>hardware</category>
      <business_duration />
      <incident_state>2</incident_state>
      <activity_due>2016-01-22 16:14:54</activity_due>
      <correlation_display />
      <company>
         <link>https://instance.service-now.com/api/now/table/core_company/31bea3d53790200044e0bfc8bcbe5dec</link>
         <value>31bea3d53790200044e0bfc8bcbe5dec</value>
      </company>
      <active>true</active>
      <due_date />
      <assignment_group>
Can't access Exchange server - is it down?

© 2017 ServiceNow. All rights reserved. 3085
Table API - PATCH /now/table/{tableName}/{sys_id}
This method updates the specified record with the request body.

URL format

Versioned URL: /api/now/v2/table/{tableName}/{sys_id}
Default URL: /api/now/table/{tableName}/{sys_id}

Supported request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_display_value</td>
<td>Data retrieval operation for reference and choice fields. Based on this value, retrieves the display value and/or the actual value from the database.</td>
</tr>
<tr>
<td></td>
<td>• true returns display values for all fields.</td>
</tr>
<tr>
<td></td>
<td>• false returns actual values from the database.</td>
</tr>
<tr>
<td></td>
<td>• If a value is not specified, this parameter defaults to false.</td>
</tr>
<tr>
<td></td>
<td>• all returns both actual and display values.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> There is no preferred method for setting this parameter. However, specifying the display value may cause performance issues since it is not reading directly from the database and may include referencing other fields and records. For more information on display values and actual values, see Table API FAQs (KB0534905).</td>
</tr>
<tr>
<td>sysparm_fields</td>
<td>Comma-separated field names to return in the response.</td>
</tr>
<tr>
<td>sysparm_view</td>
<td>UI view to determine fields returned in the response.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If both sysparm_fields and sysparm_view are specified, the sysparm_fields parameter takes priority.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sysparm_input_display_value</td>
<td>Data insert or update operations.</td>
</tr>
<tr>
<td></td>
<td>For values in the request:</td>
</tr>
<tr>
<td></td>
<td>• true treats input values as display values and they are manipulated so</td>
</tr>
<tr>
<td></td>
<td>they can be stored properly in the database.</td>
</tr>
<tr>
<td></td>
<td>• false treats input values as actual values and stored them in the</td>
</tr>
<tr>
<td></td>
<td>database without manipulation.</td>
</tr>
</tbody>
</table>

**Note:**

• If this parameter is set to true, pay attention to input values, especially date values, as these are interpreted as being supplied via the user time zone preference and are transformed into UTC format.

• To set the value of an encrypted field, you must set this parameter to true. If this parameter is not set to true, values submitted to encrypted fields are not saved. Additionally, the requesting user must have the appropriate encryption context prior to submitting the request. Encrypted fields are hidden for users without the appropriate encryption context. For more information on display values and actual values, see Table API FAQs (KB0534905). For more information on field encryption see Use an encrypted text field on page 2532.

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

**Table 958: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-no-response-body</td>
<td>By default, responses include body content detailing the modified record.</td>
</tr>
<tr>
<td></td>
<td>Set this header to true in the request to suppress the response body.</td>
</tr>
</tbody>
</table>
**Response headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>The data format of the request body.</td>
</tr>
</tbody>
</table>

**Status codes**

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see [REST API response codes](#).

**Table 960: Status codes**

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the request completed successfully.</td>
</tr>
</tbody>
</table>

**Sample cURL request**

```bash
curl "https://instance.service-now.com/api/now/table/incident/ef43c6d40a0b5700c77f9bf387afe3" \
  --request PATCH \
  --header "Accept:application/json" \
  --header "Content-Type:application/json" \
  --data "\n  {"assigned_to":"681b365ec0a80164000fb05854a0cd","urgency":"1","comments":"Elevating urgency, this is a blocking issue"}" \n  --user 'admin':'admin'
```

```json
{
  "result": {
    "upon_approval": "proceed",
    "location": {
      "link": "https://instance.service-now.com/api/now/table/cmnc_location/108752c8c611227501d4ab0e392ba97f",
      "value": "108752c8c611227501d4ab0e392ba97f"
    },
    "expected_start": "",
    "reopen_count": "",
    "close_notes": "",
    "additional_assignee_list": "",
    "impact": "1",
    "urgency": "1",
    "correlation_id": "",
    "sys_tags": "",
    "sys_domain": {
      "link": "https://instance.service-now.com/api/now/table/sys_user_group/global",
      "value": "global"
    }
  }
}
```
Can't access Exchange server - is it down?
"close_code": "",
"sys_id": "ef43c6d40a0a0b5700c77f9bf387afe3",
"approval": "not requested",
"caused_by": "",
"severity": "3",
"sys_created_by": "glide.maint",
"resolved_at": "",
"assigned_to": {
    "link": "https://instance.service-now.com/api/now/table/sys_user/681b365ec0a80164000fb0b05854a0cd",
    "value": "681b365ec0a80164000fb0b05854a0cd"
},
"business_stc": "",
"wf_activity": "",
"sys_domain_path": "/",
"cmdb_ci": {
    "link": "https://instance.service-now.com/api/now/table/cmdb_ci/281190e3c0a8000b003f593aa3f20ca6",
    "value": "281190e3c0a8000b003f593aa3f20ca6"
},
"opened_by": {
    "link": "https://instance.service-now.com/api/now/table/sys_user/glide.maint",
    "value": "glide.maint"
},
"subcategory": "",
"rejection_goto": "",
"sys_class_name": "incident",
"watch_list": "",
"time_worked": "",
"contact_type": "phone",
"escalation": "3",
"comments": ""
}

Sample Python request

```
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/table/incident/ef43c6d40a0a0b5700c77f9bf387afe3'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type": "application/xml","Accept": "application/xml"}

# Do the HTTP request
response = requests.patch(url, auth=(user, pwd), headers=headers ,data="<request><entry><assigned_to>681b365ec0a80164000fb0b05854a0cd</assigned_to><urgency>1</urgency><comments>Elevating urgency, this is a blocking issue</comments></entry></request>"

# Check for HTTP codes other than 200
```
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:',
    response.headers, 'Error Response:',response.json())
exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)

<?xml version="1.0" encoding="UTF-8"?>
<response>
    <result>
        <uponApproval>proceed</uponApproval>
        <location>
            <link>https://instance.service-now.com/api/now/table/cmn_location/108752c8c611227501d4ab0e392ba97f</link>
            <value>108752c8c611227501d4ab0e392ba97f</value>
        </location>
        <expected_start />
        <reopen_count />
        <close_notes />
        <additional_assignee_list />
        <impact>1</impact>
        <urgency>1</urgency>
        <correlation_id />
        <priority>1</priority>
        <sys_tags />
        <sys_domain>
            <link>https://instance.service-now.com/api/now/table/sys_user_group/global</link>
            <value>global</value>
        </sys_domain>
        <description />
        <group_list />
        <delivery_plan />
        <sys_mod_count>8</sys_mod_count>
        <work_notes_list />
        <follow_up />
        <business_service />
        <closed_at />
        <sla_due>2017-07-05 05:58:24</sla_due>
        <delivery_task />
        <sys_updated_on>2016-01-22 14:14:54</sys_updated_on>
        <parent />
        <work_end />
        <number>INC0000050</number>
        <closed_by />
        <work_start />
        <calendar_stc />
        <category>hardware</category>
        <business_duration />
        <incident_state>2</incident_state>
        <activity_due>2016-01-22 16:14:54</activity_due>
        <correlation_display />
        <company>
            <link>https://instance.service-now.com/api/now/table/core_company/31bea3d537902000044e0bfc8bcbe5dec</link>
            <value>31bea3d537902000044e0bfc8bcbe5dec</value>
        </company>
        <active>true</active>
        <due_date />
        <assignment_group>
<link>https://instance.service-now.com/api/now/table/sys_user_group/8a5055c9c61122780043563ef53438e3</link>
<value>8a5055c9c61122780043563ef53438e3</value>
</assignment_group>
<caller_id>
<link>https://instance.service-now.com/api/now/table/sys_user/5b7c200d0a640069006b3845b5d0fa7c</link>
<value>5b7c200d0a640069006b3845b5d0fa7c</value>
</caller_id>

<knowledge>false</knowledge>
<made_sla>true</made_sla>
<comments_and_work_notes />
<parent_incident />
<state>2</state>
<user_input />
<sys_created_on>2015-11-02 18:05:40</sys_created_on>
<approval_set />
<reassignment_count>0</reassignment_count>
<rfc />
<opened_at>2015-11-02 21:58:24</opened_at>
<child_incidents />
<order />
<short_description>Can't access Exchange server - is it down?</short_description>

<resolved_by />
<sys_updated_by>admin</sys_updated_by>
<notify>1</notify>
<upon_reject>cancel</upon_reject>
<approval_history />
<problem_id />
<calendar_duration />
<work_notes />
<close_code />
<approval>not requested</approval>
<sys_id>ef43c6d40a0a0b5700c77f9bf387afe3</sys_id>
<caused_by />
<severity>3</severity>
<sys_created_by>glide.maint</sys_created_by>
<assigned_to>
<link>https://instance.service-now.com/api/now/table/sys_user/681b365ec0a80164000fb0b05854a0cd</link>
<value>681b365ec0a80164000fb0b05854a0cd</value>
</assigned_to>
<resolved_at />
<business_stc />
<cmdb_ci>
<link>https://instance.service-now.com/api/now/table/cmdb_ci/281190e3c0a8000b003f593aa3f20ca6</link>
<value>281190e3c0a8000b003f593aa3f20ca6</value>
</cmdb_ci>
</sys_domain_path/>
<wf_activity />
<opened_by>
<link>https://instance.service-now.com/api/now/table/sys_user/glide.maint</link>
<value>glide.maint</value>
</opened_by>
<rejection_goto />
<subcategory />
<sys_class_name>incident</sys_class_name>
<watch_list />
<escalation>3</escalation>
<contact_type>phone</contact_type>
<time_worked />
Table API - DELETE /now/table/{tableName}/{sys_id}
This method deletes the specified record from the specified table.

**URL format**

Versioned URL: /api/now/v2/table/{tableName}/{sys_id}
Default URL: /api/now/table/{tableName}/{sys_id}

**Supported request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see **REST API headers**.

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**Table 963: Response headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**Status codes**

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see **REST API response codes**.
<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>Indicates the request completed successfully.</td>
</tr>
</tbody>
</table>

Sample cURL request
```bash
curl "https://instance.service-now.com/api/now/table/incident/d977b66a4f411200adf9f8e18110c7b2" \
  --request DELETE \
  --header "Accept:application/json" \
  --user 'admin':'admin'
```

Sample Python request
```python
#Need to install requests package for python
#easy_install requests
import requests
# Set the request parameters
url = 'https://instance.service-now.com/api/now/table/incident/d977b66a4f411200adf9f8e18110c7b2'
# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'
# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}
# Do the HTTP request
response = requests.delete(url, auth=(user, pwd),
  headers=headers )
# Check for HTTP codes other than 200
if response.status_code != 200:
  print('Status:', response.status_code, 'Headers:',
  response.headers, 'Error Response:',response.json())
  exit()
# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
None
```

Table API security
Access to tables via the REST API is restricted by BasicAuth.
ACLs defined for tables are enforced to restrict access to data.
Geneva

ServiceNow

ServiceNow Platform

To allow access to tables without any authentication and authorization, add the table name to
sys_public.list. ACLs defined on tables are still enforced, and it is the customer's responsibility to
deactivate ACLs on tables.
Tables protected with a role that requires elevated privileges, such as the security_admin role, are not
accessible using the Table API. Attempting to access one of these tables will return a 403 "User Not
Authorized" error.
Table API Ruby examples
Examples that demonstrate how to use the Table API with the Ruby language.

Example URLs
In the examples, replace myinstance.service-now.com with the URL of your instance.

GET (Specific Record)
#!/usr/bin/env ruby
require 'base64'
# https://rubygems.org/gems/rest-client
# Example install using gem
#
gem install rest-client
require 'rest-client'
# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'c5d0db50eb2211004d7763fba206fe13'
begin # Get the incident with sys_id declared above
response = RestClient. get ( "#{host}/api/now/table/incident/#{sys_id}",
{:authorization => "Basic
#{Base64.strict_encode64(" #{user}:#{pwd}")}", :accept => 'application/
json' } ) puts "#{response.to_str}" puts "Response status: #{response.code}"
response. headers. each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
puts "ERROR: #{e}" end

GET (All Records)
#!/usr/bin/env ruby
require 'base64'
# https://rubygems.org/gems/rest-client # Example install using gem #
gem install rest-client require 'rest-client'
# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'

©

2017 ServiceNow. All rights reserved.

3095


begin # Get ALL incidents
  response = RestClient.get("#{host}/api/now/table/incident",
    {:authorization => "Basic #{Base64.strict_encode64(" #{user}:#{pwd})}", :accept => 'application/json' }
  ) puts "#{response.to_str}"
  puts "Response status: #{response.code}"
  response. headers. each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
  puts "ERROR: #{e}" end

POST

#!/usr/bin/env ruby

require 'base64'

# https://rubygems.org/gems/json # Example install using gem
#   gem install json
require 'json'

# https://rubygems.org/gems/rest-client # Example install using gem
#   gem install rest-client
require 'rest-client'

# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'

request_body_map = { :active => 'false',
  :business_stc => '13203',
  :calendar_stc => '86415',
  :category => 'network',
  :description => 'User cannot access email on mail.company.com.',
  :escalation => '0',
  :impact => '1',
  :incident_state => '7',
  :knowledge => 'false',
  :made_sla => 'false',
  :notify => '1',
  :number => 'INC0000001',
  :priority => '2',
  :severity => '1',
  :short_description => 'Cannot read email',
  :state => '7',
  :sys_class_name => 'incident',
  :sys_created_by => 'pat',
  :sys_mod_count => '19',
  :sys_updated_by => 'glide.maint',
  :urgency => '1',
}

begin
  response = RestClient.post("#{host}/api/now/table/incident",
    request_body_map.to_json,
    { :authorization => "Basic #{Base64.strict_encode64(" #{user}:#{pwd})}",
      :content_type => "application/json",
      :accept => 'application/json' }
  ) puts "#{response.to_str}"
  puts "Response status: #{response.code}"
  response. headers. each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
  puts "ERROR: #{e}" end
PUT

#!/usr/bin/env ruby

require 'base64'

# https://rubygems.org/gems/json # Example install using gem # gem install json require 'json'

# https://rubygems.org/gems/rest-client # Example install using gem # gem install rest-client require 'rest-client'

# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'd71da88ac0a801670061eabfe4b28f77'

begin # First GET the record
  response = RestClient.get("#{host}/api/now/table/incident/#{sys_id}",
                          {:authorization => "Basic #{Base64.strict_encode64("#{user}:#{pwd}\"", :accept => 'application/json')}, :accept => 'application/json'
                            if response.code != 200
                  puts "#{response.to_str}"
                  puts "Response status: #{response.code}"
                response.headers.each { |k,v | puts "Header: #{k}=#{v}" }
        abort "GET request failed" end

  # decode the response
  result_hash = JSON.parse(response.to_str)
  incident_details = result_hash['result']

  # Update the incident record --- Let's prepend 'Updated via REST! ' to
  # the short_description field
  incident_details['short_description'] = "Updated via REST! #{incident_details['short_description']}"

  # PUT the entire record to update it
  response = RestClient.put("#{host}/api/now/table/incident/#{sys_id}",
                          incident_details.to_json,
                          {:cookies => response.cookies, #:content_type => 'application/json', :accept => 'application/json'} )

  rescue => e
        puts "ERROR: #{e}" end

PATCH

#!/usr/bin/env ruby

require 'base64'

# https://rubygems.org/gems/json # Example install using gem # gem install json require 'json'
# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'd71da88ac0a801670061eabfe4b28f77'

request_body_map = { :short_description => 'New short description', }

begin # PATCH the record with just the field you want to update
response = RestClient.patch("#{host}/api/now/table/incident/#{sys_id}",
  request_body_map.to_json, 
  Encode the short_description as JSON (:authorization => "Basic
  #{Base64.strict_encode64("#{user}:#(pwd)")}", :content_type =>
  'application/json',
  :accept => 'application/json')
) puts
"#{response.to_str}" puts "Response status: #{response.code}"
response.headers.each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
  puts "ERROR: #{e}" end

DELETE

#!/usr/bin/env ruby
require 'base64'

# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'd71da88ac0a801670061eabfe4b28f77'

begin # Delete the incident with sys_id declared above
response = RestClient.delete("#{host}/api/now/table/incident/#{sys_id}",
  {:authorization => "Basic
  #{Base64.strict_encode64("#{user}:#(pwd)")}", :accept => 'application/json' }
) puts
"#{response.to_str}" puts "Response status: #{response.code}"
response.headers.each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
  puts "ERROR: #{e}" end

Table API language examples
Review examples to understand how to access the Table API using various programming languages.

Note: The samples on this page are provided to get you started and may require changes per your own requirements and environments. ServiceNow does not, however, provide support if you customize these scripts.
Table API Perl examples
Examples that demonstrate how to use the Table API with the Perl language.

**Example URLs**

In the examples, replace `myinstance.service-now.com` with the URL of your instance.

**GET (Specific Record)**

```perl
#!/usr/bin/env perl -w
use strict;
use warnings;
use MIME::Base64;

# http://search.cpan.org/~mcrawfor/REST-Client/lib/REST/Client.pm
# Example install using cpanm:
#   sudo cpanm -i REST::Client
use REST::Client;

# Set the request parameters
my $host = 'https://myinstance.service-now.com';
my $user = 'admin';
my $pwd = 'admin';
my $sys_id = '0818562ca8d31100a92e8545569edcb0';
my $client = REST::Client->new(host => $host);
my $encoded_auth = encode_base64("$user:$pwd", '');

# Get the incident with sys_id declared above
$client->GET("/api/now/table/incident/$sys_id",
    {'Authorization' => "Basic $encoded_auth",
     'Accept' => 'application/json'});

print 'Response: ' . $client->responseContent() . "\n";
print 'Response status: ' . $client->responseCode() . "\n";
foreach ( $client->responseHeaders() ) {
    print 'Header: ' . $_ . '=' . $client->responseHeader($_) . "\n";
}
```

**GET (All Records in Table)**

```perl
#!/usr/bin/env perl -w
use strict;
use warnings;
use MIME::Base64;

# http://search.cpan.org/~mcrawfor/REST-Client/lib/REST/Client.pm
# Example install using cpanm:
#   sudo cpanm -i REST::Client
use REST::Client;

# Set the request parameters
my $host = 'https://myinstance.service-now.com';
my $user = 'admin';
my $pwd = 'admin';
my $client = REST::Client->new(host => $host);
```
post

#!/usr/bin/env perl -w
use strict;
use warnings;
use MIME::Base64;

# http://search.cpan.org/~makamaka/JSON/lib/JSON.pm
# Example install using cpanm:
#   sudo cpanm -i JSON
use JSON;

# http://search.cpan.org/~mcrawfor/REST-Client/lib/REST/Client.pm
# Example install using cpanm:
#   sudo cpanm -i REST::Client
use REST::Client;

# Set the request parameters
my $host = 'https://myinstance.service-now.com';
my $user = 'admin';
my $pwd = 'admin';

my $request_body_map = (
  'active' => 'false',
  'business_stc' => '13203',
  'calendar_stc' => '86415',
  'category' => 'network',
  'description' => 'User cannot access email on mail.company.com.',
  'escalation' => '0',
  'impact' => '1',
  'incident_state' => '7',
  'knowledge' => 'false',
  'made_sla' => 'false',
  'notify' => '1',
  'number' => 'INC0000001',
  'priority' => '2',
  'severity' => '1',
  'short_description' => 'Cannot read email',
  'state' => '7',
  'sys_class_name' => 'incident',
  'sys_created_by' => 'pat',
  'sys_mod_count' => '19',
  'sys_updated_by' => 'glide.maint',
  'urgency' => '1',
);

my $client = REST::Client->new(host => $host);

my $encoded_auth = encode_base64("$user:$pwd", '');

# Get ALL incidents
$client->GET('/api/now/table/incident',
  { 'Authorization' => "Basic $encoded_auth",
    'Accept' => 'application/json' });

print 'Response: ' . $client->responseContent() . "\n";
print 'Response status: ' . $client->responseCode() . "\n";
foreach ( $client->responseHeaders() ) {
  print 'Header: ' . $_ . '=' . $client->responseHeader($_) . "\n";
}

POST
my $encoded_auth = encode_base64("$user:$pwd", '');

# POST to the incident table
$client->POST('/api/now/table/incident',
    encode_json($%request_body_map),
    {'Authorization' => "Basic $encoded_auth",
     'Content-Type' => 'application/json',
     'Accept' => 'application/json'});

print 'Response: ' . $client->responseContent() . "\n";
print 'Response status: ' . $client->responseCode() . "\n";
foreach ( $client->responseHeaders() ) {
    print 'Header: ' . $_ . '=' . $client->responseHeader($_) . "\n";
}

PUT

#!/usr/bin/env perl -w
use strict;
use warnings;
use MIME::Base64;

# http://search.cpan.org/~makamaka/JSON/lib/JSON.pm
# Example install using cpanm:
#   sudo cpanm -i JSON
use JSON;

# http://search.cpan.org/~mcrawfor/REST-Client/lib/REST/Client.pm
# Example install using cpanm:
#   sudo cpanm -i REST::Client
use REST::Client;

# Set the request parameters
my $host = 'https://myinstance.service-now.com';
my $user = 'admin';
my $pwd = 'admin';
my $sys_id = '447547e0a8571100a92e8545569edc90';
my $encoded_auth = encode_base64("$user:$pwd", '');

my $client = REST::Client->new(host => $host);

# First GET the record
$client->GET('/api/now/table/incident/$sys_id',
    {'Authorization' => "Basic $encoded_auth",
     'Accept' => 'application/json'});

if ($client->responseCode() ne '200') {
    print 'Response: ' . $client->responseContent() . "\n";
    print 'Status: ' . $client->responseCode() . "\n";
    foreach ( $client->responseHeaders() ) {
        print 'Header: ' . $_ . '=' . $client->responseHeader($_) . "\n";
    }
    die('GET request failed');
}

# decode the response
my $result_hash = decode_json($client->responseContent());
my $incident_details = @{$result_hash}{'result'};
# Update the incident record --- Let's prepend 'Updated via REST!' to the short_description field
$incident_details->{'short_description'} = 'Updated via REST! '.

# re-encode the incident as JSON, and PUT the entire record to update it
$client->PUT("/api/now/table/incident/$sys_id",
encode_json($incident_details),
{ 'Authorization' => "Basic $encoded_auth",
  'Content-Type' => 'application/json',
  'Accept' => 'application/json' });

print 'Response: ' . $client->responseContent() . "\n";
print 'Status: ' . $client->responseCode() . "\n";
foreach ( $client->responseHeaders() ) {
  print 'Header: ' . $_ . '=' . $client->responseHeader($_) . "\n";
}

PATCH

#!/usr/bin/env perl -w
use strict;
use warnings;
use MIME::Base64;

# http://search.cpan.org/~makamaka/JSON/lib/JSON.pm
# Example install using cpanm:
#   sudo cpanm -i JSON
use JSON;

# http://search.cpan.org/~mcrawfor/REST-Client/lib/REST/Client.pm
# Example install using cpanm:
#   sudo cpanm -i REST::Client
use REST::Client;

# Set the request parameters
my $host = 'https://myinstance.service-now.com';
my $user = 'admin';
my $pwd = 'admin';
my $sys_id = '0818562ca8d31100a92e8545569edcb0';

my $request_body_map = {
  'short_description' => 'New short description'
};

my $client = REST::Client->new(host => $host);

my $encoded_auth = encode_base64("$user:$pwd", '');

# PATCH the record with just the field(s) you want to update
$client->PATCH("/api/now/table/incident/$sys_id",
  encode_json($request_body_map),
  { 'Authorization' => "Basic $encoded_auth",
    'Content-Type' => 'application/json',
    'Accept' => 'application/json' });

print 'Response: ' . $client->responseContent() . "\n";
print 'Response status: ' . $client->responseCode() . "\n";
foreach ( $client->responseHeaders() ) {
  print 'Header: ' . $_ . '=' . $client->responseHeader($_) . "\n";
}
Geneva

ServiceNow

ServiceNow Platform

}

DELETE
#!/usr/bin/env perl -w
use strict;
use warnings;
use MIME::Base64;
# http://search.cpan.org/~mcrawfor/REST-Client/lib/REST/Client.pm
# Example install using cpanm:
#
sudo cpanm -i REST::Client
use REST::Client;
# Set the request parameters
my $host = 'https://myinstance.service-now.com';
my $user = 'admin';
my $pwd = 'admin';
my $sys_id = 'a5ce3620a8571100a92e8545569edc99';
my $client = REST::Client->new(host => $host);
my $encoded_auth = encode_base64("$user:$pwd", '');
# Delete the incident with sys_id declared above
$client->DELETE("/api/now/table/incident/$sys_id",
{'Authorization' => "Basic $encoded_auth",
'Accept' => 'application/json'});
print 'Response: ' . $client->responseContent() . "\n";
print 'Status: ' . $client->responseCode() . "\n";
foreach ( $client->responseHeaders() ) {
print 'Header: ' . $_ . '=' . $client->responseHeader($_) . "\n";
}
Table API Python examples
Examples that demonstrate how to use the Table API with the Python language.

Example URLs
In the examples, replace myinstance.service-now.com with the URL of your instance.

GET
#Need to install requests package for python
#sudo easy_install requests
import requests
# Set the request parameters
url = 'https://myinstance.service-now.com/api/now/table/incident?
sysparm_limit=10'
user = 'admin'
pwd = 'admin'
# Set proper headers
headers = {"Accept":"application/json"}

©

2017 ServiceNow. All rights reserved.

3103


Geneva

ServiceNow

ServiceNow Platform

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers )
# Check for HTTP codes other than 200
if response.status_code != 200:
print('Status:', response.status_code, 'Headers:', response.headers,
'Error Response:',response.json())
exit()

# Decode the JSON response into a dictionary and use the data
print('Status:',response.status_code,'Headers:',response.headers,'Response:',response.j
print('Cookies', response.cookies)

POST
#Need to install requests package for python
#sudo easy_install requests
import requests
# Set the request parameters
url = 'https://myinstance.service-now.com/api/now/table/incident'
user = 'admin'
pwd = 'admin'
# Set proper headers
headers = {"Content-Type":"application/json","Accept":"application/json"}
# Do the HTTP request
response = requests.post(url, auth=(user, pwd),
headers=headers ,data='{"short_description":"Test"}')
# Check for HTTP codes other than 200
if response.status_code != 201:
print('Status:', response.status_code, 'Headers:', response.headers,
'Error Response:',response.json())
exit()
# Decode the JSON response into a dictionary and use the data

print('Status:',response.status_code,'Headers:',response.headers,'Response:',response.j

PUT
#Need to install requests package for python
#sudo easy_install requests
import requests
# Set the request parameters
url = 'https://myinstance.service-now.com/api/now/table/
incident/7ca3ef46d4b61100a92e188eafa1ddcd'
user = 'admin'
pwd = 'admin'
# Set proper headers
headers = {"Content-Type":"application/json","Accept":"application/json"}
# Do the HTTP request

©

2017 ServiceNow. All rights reserved.

3104


Geneva

ServiceNow

ServiceNow Platform

response = requests.put(url, auth=(user, pwd),
headers=headers ,data='{"short_description":"Test Update"}')
# Check for HTTP codes other than 200
if response.status_code != 200:
print('Status:', response.status_code, 'Headers:', response.headers,
'Error Response:',response.json())
exit()

# Decode the JSON response into a dictionary and use the data
print('Status:',response.status_code,'Headers:',response.headers,'Response:',response.j

PATCH
#Need to install requests package for python
#sudo easy_install requests
import requests
# Set the request parameters
url = 'https://myinstance.service-now.com/api/now/table/
incident/7ca3ef46d4b61100a92e188eafa1ddcd'
user = 'admin'
pwd = 'admin'
# Set proper headers
headers = {"Content-Type":"application/json","Accept":"application/json"}
# Do the HTTP request
response = requests.patch(url, auth=(user, pwd),
headers=headers ,data='{"short_description":"Test update Patch"}')
# Check for HTTP codes other than 200
if response.status_code != 200:
print('Status:', response.status_code, 'Headers:', response.headers,
'Error Response:',response.json())
exit()

# Decode the JSON response into a dictionary and use the data
print('Status:',response.status_code,'Headers:',response.headers,'Response:',response.j

DELETE
#Need to install requests package for python
#sudo easy_install requests
import requests
# Set the request parameters
url = 'https://myinstance.service-now.com/api/now/table/incident/
b684ebc6d4b61100a92e188eafa1dd65'
user = 'admin'
pwd = 'admin'
# Set proper headers
headers = {"Content-Type":"application/json","Accept":"application/json"}
# Do the HTTP request
response = requests.delete(url, auth=(user, pwd), headers=headers)

©

2017 ServiceNow. All rights reserved.

3105


# Check for HTTP codes other than 200
if response.status_code != 204:
    print('Status:', response.status_code, 'Headers:', response.headers,
    'Error Response:',response.json())
exit()

# Decode the JSON response into a dictionary and use the data
print('Status:',response.status_code,'Headers:',response.headers)

---

Table API server-side JavaScript examples

Examples that demonstrate how to use the Table API with the server-side JavaScript.

**Example URLs**

In the examples, replace `myinstance.service-now.com` with the URL of your instance.

**GET**

```javascript
var request = new GlideHTTPRequest
    ('https://myinstance.service-now.com/api/now/table/incident');
request.setBasicAuth("admin","admin");
request.addHeader('Accept','application/json');

var response = request.get();
gs.log(response.getStatusCode());
gs.log(response.getBody());
```

**POST**

```javascript
var request = new GlideHTTPRequest
    ('https://myinstance.service-now.com/api/now/table/incident');
request.setBasicAuth("admin","admin");
request.addHeader('Accept','application/json');
request.addHeader('Content-Type','application/json');

var response = request.post({"short_description":"Test me"});
gs.log(response.getStatusCode());
gs.log(response.getBody());
```

**PUT**

```javascript
var request = new GlideHTTPRequest
    ('https://myinstance.service-now.com/api/now/table/incident/0fbbeddad47e1100a92e188eafa1dd60');
request.setBasicAuth("admin","admin");
request.addHeader('Accept','application/json');
request.addHeader('Content-Type','application/json');

var response = request.put({"short_description":"Test me update"});
gs.log(response.getStatusCode());
```
PATCH

```javascript
var request = new GlideHTTPRequest
    ('https://myinstance.service-now.com/api/now/table/incident/0fbbeddad47e1100a92e188eafa1dd60');
request.setBasicAuth("admin","admin");
request.addHeader('Accept','application/json');
request.addHeader('Content-Type','application/json');
request.addHeader('X-Http-Method-Override','PATCH');

var response = request.post({"short_description":"Test me patch update"});

gs.log(response.getStatusCode());
gs.log(response.getBody());
```

DELETE

```javascript
var request = new GlideHTTPRequest
    ('https://myinstance/api/now/table/incident/0fbbeddad47e1100a92e188eafa1dd60');
request.setBasicAuth("admin","admin");
request.addHeader('Accept','application/json');
request.addHeader('Content-Type','application/json');

var response = request.del();

gs.log(response.getStatusCode());
gs.log(response.getBody());
```

Table API cURL examples

Examples that demonstrate how to use the Table API with the cURL command language.

Example URLs

In the examples, replace `myinstance.service-now.com` with the URL of your instance.

GET

```bash
curl --user admin:admin \
--header "Accept: application/json" \
https://myinstance.service-now.com/api/now/table/incident
```

POST

```bash
curl --user admin:admin \
--header "Content-Type:application/json" \n--header "Accept: application/json" \
--request POST \
```
PUT

```bash
curl --user admin:admin \
--header "Content-Type:application/json" \
--header "Accept: application/json" \
--request PUT \
--data '{"short_description":"Test with CURL"}' \
https://myinstance.service-now.com/api/now/table/incident/bf1defe3d7031100261253b2b2520373
```

PATCH

```bash
curl --user admin:admin \
--header "Content-Type:application/json" \
--header "Accept: application/json" \
--request PATCH \
--data '{"short_description":"Test with CURL Patch"}' \
https://myinstance.service-now.com/api/now/table/incident/bf1defe3d7031100261253b2b2520373
```

DELETE

```bash
curl --user admin:admin \
--header "Content-Type:application/json" \
--header "Accept: application/json" \
--request DELETE \
https://myinstance.service-now.com/api/now/table/incident/bf1defe3d7031100261253b2b2520373
```

Additional Parameters

To improve the formatting of your JSON response, add `| grep }| python -mjson.tool`.

```bash
curl -i --user admin:admin \
--header "Accept: application/json" \
https://myinstance.service-now.com/api/now/table/incident | grep }| python -mjson.tool
```

Table API Java examples

Examples that demonstrate how to use the Table API with the Java language.

For these code samples, the following is assumed:

- The project includes `apache commons-httpclient`.
- The package declaration needs to be changed accordingly.
- This is a sample code snippet that shows how to get started; it is not recommended as production-ready code.

In the examples, replace `myinstance.service-now.com` with the URL of your instance.
GET

```java
package examples.rest.java;

import java.io.IOException;
import org.apache.http.HttpException;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.util.EntityUtils;

public class GetAction {
    public static void main(String[] args) throws IOException, HttpException {
        GetAction restAction = new GetAction();
        restAction.getRequest();
    }

    public void getRequest() throws HttpException, IOException {
        CredentialsProvider credsProvider = new BasicCredentialsProvider();
        credsProvider.setCredentials(new AuthScope(new HttpHost("instance.service-now.com")),
                                   new UsernamePasswordCredentials("WebServiceUser",
                                   "WebServiceUserPassword"));
        CloseableHttpClient httpclient = HttpClients.custom().setDefaultCredentialsProvider(credsProvider).build();
        try {
            HttpGet httpget = new HttpGet("https://instance.service-now.com/api/now/table/incident");
            httpget.setHeader("Accept", "application/json");
            System.out.println("Executing request " +
            httpget.getRequestLine());
            CloseableHttpResponse response = httpclient.execute(httpget);
            try {
                System.out.println("----------------------------------------");
                System.out.println(response.getStatusLine());
                String responseBody = EntityUtils.toString(response.getEntity());
                System.out.println(responseBody);
            } finally {
                response.close();
            } finally {
                httpclient.close();
            }
        }
    }
}
```
Geneva

ServiceNow

ServiceNow Platform

POST
package examples.rest.java;
import java.io.IOException;
import
import
import
import
import
import
import
import
import
import
import
import
import

org.apache.http.HttpException;
org.apache.http.HttpHost;
org.apache.http.auth.AuthScope;
org.apache.http.auth.UsernamePasswordCredentials;
org.apache.http.client.CredentialsProvider;
org.apache.http.entity.ByteArrayEntity;
org.apache.http.impl.client.BasicCredentialsProvider;
org.apache.http.util.EntityUtils;

public class PostAction {
public static void main(String[] args) throws IOException, HttpException {
PostAction restAction = new PostAction();
restAction.postRequest();
}
public void postRequest() throws HttpException, IOException {
// This must be valid json string with valid fields and values from table
String postData = "{\"short_description\":\"Test with java post\"}";
CredentialsProvider credsProvider = new BasicCredentialsProvider();
credsProvider.setCredentials(
new AuthScope(new HttpHost("instance.service-now.com")),
new UsernamePasswordCredentials("WebServiceUser",
"WebServiceUserPassword"));
CloseableHttpClient httpclient = HttpClients.custom()
.setDefaultCredentialsProvider(credsProvider)
.build();
try {
HttpPost httpPost = new HttpPost("https://instance.service-now.com/
api/now/table/incident");
httpPost.setHeader("Accept", "application/json");
httpPost.setHeader("Content-Type", "application/json");
HttpEntity entity = new
ByteArrayEntity(postData.getBytes("utf-8"));
httpPost.setEntity(entity);
System.out.println("Executing request " +
httpPost.getRequestLine());
CloseableHttpResponse response = httpclient.execute(httpPost);
try {
System.out.println("----------------------------------------");
System.out.println(response.getStatusLine());
String responseBody =
EntityUtils.toString(response.getEntity());
System.out.println(responseBody);
} finally {
response.close();
}
} finally {
httpclient.close();

©

2017 ServiceNow. All rights reserved.

3110


package examples.rest.java;

import java.io.IOException;
import org.apache.http.HttpException;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.util.EntityUtils;

public class PutAction {
    public static void main(String[] args) throws IOException, HttpException {
        PutAction restAction = new PutAction();
        restAction.putRequest();
    }

    public void putRequest() throws HttpException, IOException {
        // This must be valid json string with valid fields and values from table
        String postData = "{"short_description":"Test with java post"}"
        CredentialsProvider credsProvider = new BasicCredentialsProvider();
        credsProvider.setCredentials(
            new AuthScope(new HttpHost("instance.service-now.com")),
            new UsernamePasswordCredentials("WebServiceUser",
                "WebServiceUserPassword"));
        CloseableHttpClient httpclient = HttpClients.custom().
            setDefaultCredentialsProvider(credsProvider).build();
        try {
                api/now/table/incident/1231231");
            httpPut.setHeader("Accept", "application/json");
            httpPut.setHeader("Content-Type", "application/json");
            HttpEntity entity = new ByteArrayEntity(postData.getBytes("utf-8");
            httpPut.setEntity(entity);
            System.out.println("Executing request " +
                httpPut.getRequestLine());
            CloseableHttpResponse response = httpclient.execute(httpPut);
            try {
                System.out.println("-----------------------------");
                System.out.println(response.getStatusLine());
                String responseBody =
                    EntityUtils.toestring(response.getEntity());
            } finally {
                httpclient.close();
            }
        } finally {
            httpclient.close();
        }
    }
}

package examples.rest.java;

import java.io.IOException;
import org.apache.http.HttpException;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.util.EntityUtils;

public class PatchAction {
    public static void main(String[] args) throws IOException, HttpException {
        PatchAction restAction = new PatchAction();
        restAction.patchRequest();
    }

    public void patchRequest() throws HttpException, IOException {
        // This must be valid json string with valid fields and values from table
        String postData = "{"short_description":"Test with java post"};
        CredentialsProvider credsProvider = new BasicCredentialsProvider();
        credsProvider.setCredentials(
                new AuthScope(new HttpHost("instance.service-now.com")),
                new UsernamePasswordCredentials("WebServiceUser",
                        "WebServiceUserPassword"));
        CloseableHttpClient httpclient = HttpClients.custom()
                .setDefaultCredentialsProvider(credsProvider)
                .build();
        try {
            HttpPatch httpPatch = new HttpPatch("https://instance.service-
                    now.com/api/now/table/incident/acdc4303d722310020c153b2b2520369");
            httpPatch.setHeader("Accept", "application/json");
            httpPatch.setHeader("Content-Type", "application/json");
            HttpEntity entity = new
                    ByteArrayEntity(postData.getBytes("utf-8"));
            httpPatch.setEntity(entity);
            System.out.println("Executing request "+
                    httpPatch.getRequestLine());
            CloseableHttpResponse response = httpclient.execute(httpPatch);
            System.out.println(responseBody);
        } finally {
            response.close();
        }
    }
}

PATCH
try {
    System.out.println("----------------------------------------");
    System.out.println(response.getStatusLine());
    String responseBody = EntityUtils.toString(response.getEntity());
    System.out.println(responseBody);
} finally {
    response.close();
}
} finally {
    httpClient.close();
}

package examples.rest.java;

import java.io.IOException;
import org.apache.http.HttpException;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.util.EntityUtils;

public class DeleteAction {
    public static void main(String[] args) throws IOException, HttpException {
        DeleteAction restAction = new DeleteAction();
        restAction.deleteRequest();
    }

    public void deleteRequest() throws HttpException, IOException {
        // This must be valid json string with valid fields and values from table

        CredentialsProvider credsProvider = new BasicCredentialsProvider();
        credsProvider.setCredentials(
                new AuthScope(new HttpHost("instance.service-now.com")),
                new UsernamePasswordCredentials("WebServiceUser",
                        "WebServiceUserPassword"));
        CloseableHttpClient httpClient = HttpClientBuilder.
                        custom().setDefaultCredentialsProvider(credsProvider).
                        build();

        try {
            HttpDelete httpDelete = new HttpDelete("https://instance.service-
                    now.com/api/nw/table/issue/acdc4303d722310020c153b2b2520369");
            httpDelete.setHeader("Accept", "application/json");

            System.out.println("Executing request " +
                    httpDelete.getmRequestLine());
            CloseableHttpResponse response = httpClient.execute(httpDelete);
    
        }
}

DELETE
try {
    System.out.println("----------------------------------------");
    System.out.println(response.getStatusLine());
} finally {
    response.close();
}
}
}
}
}
}

Table API JavaScript examples
Examples that demonstrate how to use the Table API with the JavaScript language.

**JavaScript Examples**

**GET**

```javascript
function getAction()
{
    var client=new XMLHttpRequest();
    client.open("GET","/api/now/table/incident");
    client.setRequestHeader('Accept','application/json');
    client.onreadystatechange = function(){
        if(this.readyState == this.DONE){
            document.getElementById("response").innerHTML=this.status + this.response;
        }
    };
    client.send();
}
```

**POST**

```javascript
function postAction()
{
    var requestBody = '{"short_description":"Test insert"}';
    var client=new XMLHttpRequest();
    client.open("POST","/api/now/table/incident");
    client.setRequestHeader('Accept','application/json');
    client.setRequestHeader('Content-Type','application/json');
    client.onreadystatechange = function(){
        if(this.readyState == this.DONE){
            document.getElementById("response").innerHTML=this.status + this.response;
        }
    };
    client.send(requestBody);
}
```
PUT

```javascript
function putAction()
{
    var requestBody = '{"short_description":"Test update"}';
    var client=new XMLHttpRequest();
    client.open("PUT","/api/now/table/incident/70a503b1f1971100a92e33bc2fbe836b");
    client.setRequestHeader('Accept','application/json');
    client.setRequestHeader('Content-Type','application/json');
    client.onreadystatechange = function(){
        if(this.readyState == this.DONE){
            document.getElementById("response").innerHTML=this.status +
            this.response;
        }};
    client.send(requestBody);
}
```

PATCH

```javascript
function patchAction()
{
    var requestBody = '{"short_description":"Test update with Patch"}';
    var client=new XMLHttpRequest();
    client.open("PATCH","/api/now/table/incident/70a503b1f1971100a92e33bc2fbe836b");
    client.setRequestHeader('Accept','application/json');
    client.setRequestHeader('Content-Type','application/json');
    client.onreadystatechange = function(){
        if(this.readyState == this.DONE){
            document.getElementById("response").innerHTML=this.status +
            this.response;
        }};
    client.send(requestBody);
}
```

DELETE

```javascript
function deleteAction()
{
    var client=new XMLHttpRequest();
    client.open("DELETE","/api/now/table/incident/70a503b1f1971100a92e33bc2fbe836b");
    client.setRequestHeader('Accept','application/json');
    client.setRequestHeader('Content-Type','application/json');
    client.onreadystatechange = function(){
        if(this.readyState == this.DONE){
            document.getElementById("response").innerHTML=this.status +
            this.response;
        }};
    client.send();
}
```
Geneva

ServiceNow

ServiceNow Platform

Table API Ruby examples
Examples that demonstrate how to use the Table API with the Ruby language.

Example URLs
In the examples, replace myinstance.service-now.com with the URL of your instance.

GET (Specific Record)
#!/usr/bin/env ruby
require 'base64'
# https://rubygems.org/gems/rest-client
# Example install using gem
#
gem install rest-client
require 'rest-client'
# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'c5d0db50eb2211004d7763fba206fe13'
begin # Get the incident with sys_id declared above
response = RestClient. get ( "#{host}/api/now/table/incident/#{sys_id}",
{:authorization => "Basic
#{Base64.strict_encode64(" #{user}:#{pwd}")}", :accept => 'application/
json' } ) puts "#{response.to_str}" puts "Response status: #{response.code}"
response. headers. each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
puts "ERROR: #{e}" end

GET (All Records)
#!/usr/bin/env ruby
require 'base64'
# https://rubygems.org/gems/rest-client # Example install using gem #
gem install rest-client require 'rest-client'
# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
begin # Get ALL incidents
response = RestClient. get ( "#{host}/api/now/table/incident",
{:authorization => "Basic
#{Base64.strict_encode64(" #{user}:#{pwd}")}", :accept => 'application/
json' } ) puts "#{response.to_str}" puts "Response status: #{response.code}"
response. headers. each { |k,v | puts "Header: #{k}=#{v}" }

©

2017 ServiceNow. All rights reserved.

3116


POST

#!/usr/bin/env ruby

  rescue => e
  puts "ERROR: #{e}" end

PUT

#!/usr/bin/env ruby

  rescue => e
  puts "ERROR: #{e}" end
require 'base64'

# https://rubygems.org/gems/json # Example install using gem # gem install json require 'json'

# https://rubygems.org/gems/rest-client # Example install using gem # gem install rest-client require 'rest-client'

# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'd71da88ac0a801670061eabfe4b28f77'

begin
  response = RestClient.get("#{host}/api/now/table/incident/#{sys_id}",
    {:
      :authorization => "Basic
      #{Base64.strict_encode64(" #{user}:#{pwd}")}",
      :accept => 'application/json'
    }
  )
  if response.code != 200
    puts "#{response.to_str}"
    puts "Response status: #{response.code}"
    abort "GET request failed"
  end

  # decode the response
  result_hash = JSON.parse(response.to_str)
  incident_details = result_hash['result']

  # Update the incident record --- Let's prepend 'Updated via REST!' to the short_description field
  incident_details['short_description'] = "Updated via REST! #{incident_details['short_description']}"

  # PUT the entire record to update it
  response = RestClient.put("#{host}/api/now/table/incident/#{sys_id}",
    incident_details.to_json,
    {:
      :cookies => response.cookies,  # Note the re-use of cookies
      :content_type => 'application/json',
      :accept => 'application/json'
    }
  )

  rescue => e
    puts "ERROR: #{e}" end

PATCH

#!/usr/bin/env ruby
require 'base64'

# https://rubygems.org/gems/json # Example install using gem # gem install json require 'json'

# https://rubygems.org/gems/rest-client # Example install using gem # gem install rest-client require 'rest-client'

# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'd71da88ac0a801670061eabfe4b28f77'

request_body_map = { :short_description => 'New short description', }

begin # PATCH the record with just the field you want to update
  response = RestClient. patch( "#{host}/api/now/table/incident/
  #{sys_id}"
    request_body_map.to_json, #
  Encode the short_description as JSON {:authorization => "Basic
  #{Base64.strict_encode64(" #{user}:#{pwd})"), :content_type =>
  'application/json',
  :accept => 'application/json' } ) puts
"#{response.to_str}" puts "Response status: #{response.code}"
  response. headers. each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
  puts "ERROR: #{e}" end

DELETE

#!/usr/bin/env ruby

require 'base64'

# https://rubygems.org/gems/rest-client # Example install using gem#
gem install rest-client require 'rest-client'

# Set the request parameters
host = 'https://myinstance.service-now.com'
user = 'admin'
pwd = 'admin'
sys_id = 'd71da88ac0a801670061eabfe4b28f77'

begin # Delete the incident with sys_id declared above
  response = RestClient. delete( "#{host}/api/now/table/incident/
  #{sys_id}"
    {:authorization => "Basic
  #{Base64.strict_encode64(" #{user}:#{pwd})"), :accept => 'application/
  json' } ) puts "#{response.to_str}" puts "Response status: #{response.code}"
  response. headers. each { |k,v | puts "Header: #{k}=#{v}" }
rescue => e
  puts "ERROR: #{e}" end

Aggregate API

The Aggregate API allows you to compute aggregate statistics about existing table and column data.

For Aggregate API requests, you must have read access for all records in the table you query. If an ACL
prevents the requesting user from accessing any record in the table, the request returns a 403 Forbidden
error.

Aggregate API - GET /now/stats/(tableName)

This method retrieves records for the specified table and performs aggregate functions on the returned
values.

URL format

Versioned URL: /api/now/v1/stats/(tableName)
Default URL: /api/now/stats/{tableName}

## Supported request parameters

### Table 965: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| sysparm_query  | An encoded query. For example: 
  
  (sysparm_query=active=true)  
  (sysparm_query=caller_id=javascript:gs.getUserID()^active=true)  
  
  If part of the query is invalid, such as by specifying an invalid field name, the instance ignores the invalid part. It then returns rows using only the valid portion of the query. You can control this behavior using the property glide.invalid_query.returns_no_rows. Set this property to true to return no rows on an invalid query. |
<p>| sysparm_group_by | Fields by which to group the returned data. You can specify multiple fields by separating each field with a comma, such as sysparm_group_by=priority, state. |
| sysparm_having | Additional query that enables you to filter the data based on an aggregate operation. The value for this parameter must follow the syntax aggregate^field^operator^value, such as count^priority^&gt;^3 to obtain the number of records within the query results with a priority greater than 3. You can specify multiple queries by separating each with a comma, such as count^state/^=1, avg^priority^&gt;^3. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_&lt;aggregate&gt;_fields</td>
<td>List of fields on which to perform each aggregate operation. You can specify multiple fields by separating each with a comma. For example, to get the average values from the duration and priority fields, use sysparm_avg_fields=duration,priority.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Specify this parameter, the sysparm_count parameter, or both for your query to return meaningful results. If neither parameter is passed, no aggregate operation is performed.</td>
</tr>
<tr>
<td>sysparm_count</td>
<td>Flag that determines whether to return the number of records returned by the query.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Specify this parameter, the sysparm_&lt;aggregate&gt;_fields parameter, or both for your query to return meaningful results. If neither parameter is passed, no aggregate operation is performed.</td>
</tr>
<tr>
<td>sysparm_display_value</td>
<td>Data retrieval operation when grouping by reference or choice fields. Based on this value, the query returns either the display value, the actual value in the database, or both.</td>
</tr>
<tr>
<td></td>
<td>- true returns display values for all fields.</td>
</tr>
<tr>
<td></td>
<td>- false returns actual values from the database. If a value is not specified, this parameter defaults to false.</td>
</tr>
<tr>
<td></td>
<td>- all returns both actual and display values.</td>
</tr>
<tr>
<td></td>
<td>There is no preferred method for setting this parameter. However, specifying the display value may cause performance issues as they are not read from the database and may reference other fields and records. For more information on display values and actual values, see Table API FAQs (KB0534905).</td>
</tr>
</tbody>
</table>
Parameter | Description
---|---
sysparm_orderby | List of values by which to order grouped results. You can specify an order using a field or an aggregate. For example, if you specify `sysparm_orderby=AVG^state`, groups of results with lower average state values are returned first. You can also order by COUNT to arrange groups of records by the number of records in each group.

When you specify an order, groups are ordered in ascending order by default. Use `^DESC` to sort in descending order, such as `sysparm_orderby=state^DESC`.

Key-Value Pairs | An alternative to using the `sysparm_query` parameter. You can filter a query using key-value pairs where the key is the name of a field.

For example, instead of using the parameter `&sysparm_query=active=true`, you can use `&active=true`. You can use the display value when the field is a choice or reference type field, such as `&state=closed` instead of `&state=7`. To specify multiple key-value pairs, separate each with an ampersand, such as `&active=true&assigned_to=john.smith`.

Headers

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see [REST API headers](#).

**Table 966: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**Table 967: Response headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see [REST API response codes](#).
Table 968: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates that the request completed successfully.</td>
</tr>
</tbody>
</table>

Available aggregate functions

You can specify which aggregate functions to perform by using either the sysparm_<aggregate>_fields parameter or sysparm_having=<aggregate>^field^operator^value parameter, substituting <aggregate> for one of these aggregate functions:

- `avg`
- `sum`
- `min`
- `max`

Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/stats/incident?sysparm_avg_fields=reassignment_count%2Cbusiness_stc&sysparm_group_by=assignment_group" \
--request GET \
--header "Accept:application/json" \
--user 'admin':'admin'
```

```json
{
    "result": [  
        {
            "stats": {
                "avg": {
                    "business_stc": "804162.7143",
                    "reassignment_count": "1.0000"
                }
            },
            "groupby_fields": [
                {
                    "value": "",
                    "field": "assignment_group"
                }
            ]
        },
        {
            "stats": {
                "avg": {
                    "business_stc": "2037371.0000",
                    "reassignment_count": "1.5000"
                }
            },
            "groupby_fields": [
                {
                    "value": "287ee6feda9fe198100ada7950d0b1b73",
                    "field": "assignment_group"
                }
            ]
        }
    ]
}
```
{  
  "stats": {  
    "avg": {  
      "business_stc": "1821488.2857",  
      "reassignment_count": "1.1111"  
    },  
  },  
  "groupby_fields": [  
    {  
      "value": "8a5055c9c61122780043563ef53438e3",  
      "field": "assignment_group"  
    }  
  ],  
},  
{  
  "stats": {  
    "avg": {  
      "business_stc": "1730322.0000",  
      "reassignment_count": "1.2500"  
    },  
  },  
  "groupby_fields": [  
    {  
      "value": "287ebd7da9fe198100f92cc8d1d2154e",  
      "field": "assignment_group"  
    }  
  ],  
},  
{  
  "stats": {  
    "avg": {  
      "business_stc": "1564478.6250",  
      "reassignment_count": "1.2500"  
    },  
  },  
  "groupby_fields": [  
    {  
      "value": "d625dcece0a8016700a222a0f7900d06",  
      "field": "assignment_group"  
    }  
  ],  
},  
{  
  "stats": {  
    "avg": {  
      "business_stc": "1512202.2500",  
      "reassignment_count": "1.1111"  
    },  
  },  
  "groupby_fields": [  
    {  
      "value": "8a4dde73c6112278017a6a4baf547aa7",  
      "field": "assignment_group"  
    }  
  ]  
}  
}
Sample Python request

```
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/stats/incident?sysparm_avg_fields=reassignment_count%2Cbusiness_stc&sysparm_group_by=assignment_group'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {'Content-Type': 'application/json', 'Accept': 'application/json'}

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:', response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>
    <stats>
      <avg>
        <business_stc>804162.7143</business_stc>
        <reassignment_count>1.0000</reassignment_count>
      </avg>
    </stats>
    <groupby_fields>
      <field>assignment_group</field>
      <value />
    </groupby_fields>
  </result>
  <result>
    <stats>
      <avg>
        <business_stc>2037371.0000</business_stc>
        <reassignment_count>1.5000</reassignment_count>
      </avg>
    </stats>
    <groupby_fields>
      <field>assignment_group</field>
      <value>287ee6fe9fe198100ada7950d0b1b73</value>
    </groupby_fields>
  </result>
  <result>
    <stats>
      <avg>
      </avg>
    </stats>
    <groupby_fields>
      <field>assignment_group</field>
      <value />
    </groupby_fields>
  </result>
</response>
```
Aggregate API examples
These examples demonstrate how to perform a REST query using cURL commands, and the data returned for each command.

Sample response formats

```json
{
  "result": {
    "stats": {
      "count": "6",
      "max": {
        "number": "INC0000053"
      }
    }
  }
}
```
Figure 801: JSON format

```xml
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>
    <stats>
      <min>
        <priority>1</priority>
        <number>INC0000005</number>
      </min>
      <max>
        <number>INC0000050</number>
      </max>
      <count>9</count>
    </stats>
    <groupby_fields>
      <field>assignment_group</field>
      <value>Hardware</value>
    </groupby_fields>
  </result>
</response>
```
Figure 802: XML format

Specifying a query

You can specify a query by using either the sysparm_query parameter or key-value pairs.

Using sysparm_query:

curl -H "Accept:application/json" --user admin:admin
Using key-value pairs:

```curl
curl -H "Accept:application/json" --user admin:admin
    "http://<instance>.service-now.com/api/now/stats/incident?
    sysparm_count=true&sysparm_max_fields=number&sysparm_min_fields=number,priority&sysparm_query=active=true&assignment_group=software"
```

```json
{
    "result": {
        "stats": {
            "count": "6",
            "max": {
                "number": "INC0000053"
            },
            "min": {
                "priority": "1",
                "number": "INC0000015"
            }
        }
    }
}
```

**Figure 803: Response**

### Grouping results

You can group results by using the `sysparm_group_by` parameter.

```curl
curl -H "Accept:application/json" --user admin:admin
    "http://<instance>.service-now.com/api/now/stats/incident?
    sysparm_count=true&sysparm_max_fields=number&sysparm_min_fields=number,priority&sysparm_group_by=assignment_group"
```

```json
{
    "result": [ {
        "stats": {
            "count": "13",
            "max": {
                "number": "INC0000048"
            },
            "min": {
                "priority": "1",
                "number": "INC0000007"
            }
        },
        "groupby_fields": [ {
            "value": "",
            "field": "assignment_group"
        } ]
    }, {
        "stats": {
```
Figure 804: Response

Specifying a display value format

You can control if reference and choice field values appear as a sys_id, a label value, or both by using the `sysparm_display_value` parameter.

```
curl -H "Accept:application/json" --user admin:admin "http://<instance>.service-now.com/api/now/stats/incident?sysparm_count=true&sysparm_max_fields=number&sysparm_min_fields=number,priority&sysparm_group_by=assignment_group&sysparm_display_value=true"
```

```
{
    "result": [
        {
            "stats": {
                "count": "13",
                "max": {
                    "number": "INC0000048"
                },
                "min": {
                    "priority": "1",
                    "number": "INC0000007"
                }
            },
            "groupby_fields": [
                {
                    "value": "",
                    "field": "assignment_group"
                }
            ]
        }
    ]
}
```

```json
{
    "stats": {
        "count": "2",
        "max": {
            "number": "INC0000012"
        },
        "min": {
            "priority": "4",
            "number": "INC0000010"
        }
    },
    "groupby_fields": [
        {
            "value": "Database",
            "field": "assignment_group"
        }
    ]
},
{
    "stats": {
        "count": "9",
        "max": {
            "number": "INC0000050"
        },
        "min": {
            "priority": "1",
            "number": "INC0000005"
        }
    },
    "groupby_fields": [
        {
            "value": "Hardware",
            "field": "assignment_group"
        }
    ]
},
{
    "stats": {
        "count": "5",
        "max": {
            "number": "INC0000039"
        },
        "min": {
            "priority": "1",
            "number": "INC0000002"
        }
    },
    "groupby_fields": [
        {
            "value": "Network",
            "field": "assignment_group"
        }
    ]
},
{
    "stats": {
        "count": "12",
        "max": {
            "number": "INC0000055"
        },
        "min": {
            "priority": "1",
            "number": "INC0000001"
        }
    }
}
```
Ordering grouped fields

You can order groups of results based on the values in one or more specified fields by using the `sysparm_orderby` parameter.

```
curl -H "Accept:application/json" --user admin:admin
    "http://<instance>.service-now.com/api/now/stats/incident?
    sysparm_count=true&sysparm_max_fields=number&sysparm_min_fields=number,priority&sysparm_orderby=assignment_group^DESC"
```

```
{
  "result": [
    {
      "stats": {
        "count": "6",
        "max": {
          "number": "INC00000053"
        },
        "min": {
          "priority": "1",
          "number": "INC00000015"
        }
      },
      "groupby_fields": [
        {
          "value": "Software",
          "field": "assignment_group"
        }
      ]
    }
  ]
}
```


```json
] },
{
  "stats": {
    "count": "5",
    "max": {
      "number": "INC0000055"
    },
    "min": {
      "priority": "1",
      "number": "INC0000017"
    }
  },
  "groupby_fields": [
    {
      "value": "Service Desk",
      "field": "assignment_group"
    }
  ] },
{
  "stats": {
    "count": "3",
    "max": {
      "number": "INC0000039"
    },
    "min": {
      "priority": "1",
      "number": "INC0000002"
    }
  },
  "groupby_fields": [
    {
      "value": "Network",
      "field": "assignment_group"
    }
  ] },
{
  "stats": {
    "count": "4",
    "max": {
      "number": "INC0000050"
    },
    "min": {
      "priority": "1",
      "number": "INC0000005"
    }
  },
  "groupby_fields": [
    {
      "value": "Hardware",
      "field": "assignment_group"
    }
  ] },
{
  "stats": {
    "count": "12",
    "max": {
      "number": "INC0000048"
    },
    "min": {
      "priority": "1",
```
The Import Set API allows you to interact with import set tables.

The API transforms incoming data based on associated transform maps. The import set API supports synchronous transforms. The Import Set API mirrors the existing SOAP interface.

**Import Set API - POST /now/import/{tableName}**

This method inserts incoming data into a specified staging table and triggers transformation based on predefined transform maps in the import set table.

**URL format**

Versioned URL: `/api/now/v1/import/<staging table_name>`

Default URL: `/api/now/import/<staging table_name>`

**Supported request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see [REST API headers](#).

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
Table 971: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>The URL of the created resource.</td>
</tr>
</tbody>
</table>

**Status codes**

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see [REST API response codes](#).

Table 972: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Indicates the import completed successfully.</td>
</tr>
</tbody>
</table>

**Transforming inserted records**

Transformation occurs synchronously. For each transform map that you define, the responses include transformation results such as information on the target records.

**Note:** The status_message and error_message fields on transformation scripts are processed and returned in response, along with any custom response fields.

**Sample cURL request**

```bash
curl "https://instance.service-now.com/api/now/import/imp_user"
  --request POST
  --header "Accept:application/json"
  --header "Content-Type:application/json"
  --data "{"first_name':'John','last_name':'Public','user_id':'john.public','email':'john.public@company.com'}"
  --user 'admin':'admin"
```

```json
{
  "import_set": "ISET0010001",
  "staging_table": "imp_user",
  "result": [
    {
      "transform_map": "User",
      "table": "sys_user",
      "display_name": "name",
      "display_value": "John Public",
      "record_link": "https://instance.service-now.com/api/now/table/sys_user/ea928be64f411200adf9f8e18110c777",
      "status": "inserted",
      "sys_id": "ea928be64f411200adf9f8e18110c777"
    }
  ]
}
```
Geneva

ServiceNow

ServiceNow Platform

Sample Python request
#Need to install requests package for python
#easy_install requests
import requests
# Set the request parameters
url = 'https://instance.service-now.com/api/now/import/imp_user'
# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'
# Set proper headers
headers = {"Content-Type":"application/
xml","Accept":"application/xml"}
# Do the HTTP request
response = requests.post(url, auth=(user, pwd),
headers=headers ,data=" <request><entry><first_name>John</
first_name><last_name>Public</last_name><user_id>john.public</
user_id><email>john.public@company.com</email></entry></
request>")
# Check for HTTP codes other than 200
if response.status_code != 200:
print('Status:', response.status_code, 'Headers:',
response.headers, 'Error Response:',response.json())
exit()
# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
<?xml version="1.0" encoding="UTF-8"?>
<response>
<import_set>ISET0010001</import_set>
<staging_table>imp_user</staging_table>
<result>
<display_name>name</display_name>
<display_value>John Public</display_value>
<status>ignored</status>
<status_message>No field values changed</status_message>
<sys_id>ea928be64f411200adf9f8e18110c777</sys_id>
<record_link>https://instance.service-now.com/api/now/
table/sys_user/ea928be64f411200adf9f8e18110c777</record_link>
<table>sys_user</table>
<transform_map>User</transform_map>
</result>
</response>

Import Set API - GET /now/import/{tableName}/{sys_id}
This method retrieves the specified import staging record and resulting transformation result.

URL format
Versioned URL: /api/now/v1/table/{tableName}/{sys_id}
Default URL: /api/now/table/{tableName}/{sys_id}

©

2017 ServiceNow. All rights reserved.

3137


Supported request parameters

Table 973: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Headers

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

Table 974: Request headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Table 975: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see REST API response codes.

Table 976: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the query ran successfully.</td>
</tr>
<tr>
<td>404</td>
<td>Indicates the specified resource was not available. As import set tables are deleted frequently based on a schedule, GET requests may return 404 NotFound responses if the transformation result no longer exists.</td>
</tr>
</tbody>
</table>

Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/import/imp_user/e2928be64f411200adf9f8e18110c777" \
   --request GET \
```
Geneva  ServiceNow  ServiceNow Platform

```
--header "Accept:application/json" \ 
--user 'admin':'admin'

{
   "import_set": "ISET0010001",
   "staging_table": "imp_user",
   "result": [
   {uentes your
   "transform_map": "User",
   "table": "sys_user",
   "display_name": "name",
   "display_value": "John Public",
   "record_link": "https://instance.service-now.com/api/now/table/sys_user/ea928be64f411200adf9f8e18110c777",
   "status": "inserted",
   "sys_id": "ea928be64f411200adf9f8e18110c777"
   }
   ]
}
```

**Sample Python request**

```python
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/import/imp_user/e2928be64f411200adf9f8e18110c777'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:', response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <import_set>ISET0010001</import_set>
    <staging_table>imp_user</staging_table>
    <result>
        <display_name>name</display_name>
        <display_value>John Public</display_value>
        <status>inserted</status>
    </result>
</response>
```
Import Set API Examples

Use POST to import data to your instance and GET methods to query import set data.

**POST Examples**

**Table 977: Loading a record to an import set**

<table>
<thead>
<tr>
<th>Request:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; HTTP/1.1 201 Created</td>
</tr>
<tr>
<td>&lt; Location: <a href="http://https://">http://https://</a>&lt;instance name&gt;.service-now.com/api/now/import/imp_notification/127375019f23110041a496fccc67fcfe3</td>
</tr>
<tr>
<td>&lt; Content-Type: application/json</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;result&quot;: [</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;sys_id&quot;: &quot;567375019f23110041a496fccc67fcfe3&quot;,</td>
</tr>
<tr>
<td>&quot;status&quot;: &quot;inserted&quot;,</td>
</tr>
<tr>
<td>&quot;record_link&quot;:</td>
</tr>
<tr>
<td>&quot;https://&lt;instance name&gt;.service-now.com/api/now/table/incident/567375019f23110041a496fccc67fcfe3&quot;,</td>
</tr>
<tr>
<td>&quot;display_value&quot;: &quot;INC0010023&quot;,</td>
</tr>
<tr>
<td>&quot;display_name&quot;: &quot;number&quot;,</td>
</tr>
<tr>
<td>&quot;table&quot;: &quot;incident&quot;,</td>
</tr>
<tr>
<td>&quot;transform_map&quot;: &quot;Notification&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>],</td>
</tr>
<tr>
<td>&quot;staging_table&quot;: &quot;imp_notification&quot;,</td>
</tr>
<tr>
<td>&quot;import_set&quot;: &quot;ISET0010004&quot;</td>
</tr>
</tbody>
</table>

**Table 978: Results for multiple targets are returned**

| Request: |
-H "Accept: application/json" -H "Content-Type: application/json" --data '{"message":"some message", "uuid":"00005"}' -v
> POST /api/now/import/imp_notification HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> Authorization: Basic YWRtaW46YWRtaW4=
> Accept: application/json
> Content-Type: application/json
>
Response:
< HTTP/1.1 201 Created
< Location: http://https://<instance name>.service-now.com/api/now/import/imp_notification/404875019f23110041a496fcc67fcfe5
{
   "result": [
   {
      "sys_id": "484875019f23110041a496fcc67fcfe5",
      "status": "inserted",
      "record_link": "http://https://<instance name>.service-now.com/api/now/table/incident/484875019f23110041a496fcc67fcfe5",
      "display_value": "INC0010024",
      "display_name": "number",
      "table": "incident",
      "transform_map": "Notification"
   },
   {
      "sys_id": "084875019f23110041a496fcc67fcfe6",
      "status": "inserted",
      "record_link": "http://https://<instance name>.service-now.com/api/now/table/problem/084875019f23110041a496fcc67fcfe6",
      "display_value": "PRB0040025",
      "display_name": "number",
      "table": "problem",
      "transform_map": "Problem Xform"
   }
   ],
   "staging_table": "imp_notification",
   "import_set": "ISET0010004"
}
Table 979: Using a simple transform script defined on Problem Xform

<table>
<thead>
<tr>
<th>Predefined Transform Script:</th>
</tr>
</thead>
<tbody>
<tr>
<td>status_message = &quot;I’m a status message&quot;;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Request:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Response:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; HTTP/1.1 201 Created</td>
</tr>
<tr>
<td>&lt; Location: https://&lt;instance name&gt;.service-now.com/api/now/import/imp_notification/922975019f23110041a496fcc67fcf6a</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;result&quot;: [</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;sys_id&quot;: &quot;5a2975019f23110041a496fcc67fcf6a&quot;,</td>
</tr>
<tr>
<td>&quot;status&quot;: &quot;inserted&quot;,</td>
</tr>
<tr>
<td>&quot;record_link&quot;:</td>
</tr>
<tr>
<td>&quot;record_link&quot;:</td>
</tr>
<tr>
<td>&quot;<a href="http://https://">http://https://</a>&lt;instance name&gt;.service-now.com/api/now/table/incident/5a2975019f23110041a496fcc67fcf6a&quot;,</td>
</tr>
<tr>
<td>&quot;display_value&quot;: &quot;INC0010025&quot;,</td>
</tr>
<tr>
<td>&quot;display_name&quot;: &quot;number&quot;,</td>
</tr>
<tr>
<td>&quot;table&quot;: &quot;incident&quot;,</td>
</tr>
<tr>
<td>&quot;transform_map&quot;: &quot;Notification&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>,</td>
</tr>
<tr>
<td>&quot;status_message&quot;: &quot;I'm a status message&quot;, &quot;status_message&quot;: &quot;I'm a status message&quot;,</td>
</tr>
<tr>
<td>&quot;sys_id&quot;: &quot;1a2975019f23110041a496fcc67fcf6b&quot;,</td>
</tr>
<tr>
<td>&quot;status&quot;: &quot;inserted&quot;,</td>
</tr>
<tr>
<td>&quot;record_link&quot;:</td>
</tr>
<tr>
<td>&quot;record_link&quot;:</td>
</tr>
<tr>
<td>&quot;<a href="http://https://">http://https://</a>&lt;instance name&gt;.service-now.com/api/now/table/problem/1a2975019f23110041a496fcc67fcf6b&quot;,</td>
</tr>
<tr>
<td>&quot;display_value&quot;: &quot;PRB0040026&quot;,</td>
</tr>
<tr>
<td>&quot;display_name&quot;: &quot;number&quot;,</td>
</tr>
<tr>
<td>&quot;table&quot;: &quot;problem&quot;,</td>
</tr>
<tr>
<td>&quot;transform_map&quot;: &quot;Problem Xform&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>,</td>
</tr>
<tr>
<td>&quot;staging_table&quot;: &quot;imp_notification&quot;,</td>
</tr>
<tr>
<td>&quot;import_set&quot;: &quot;ISET0010004&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>
Table 980: Using a script to define custom dynamic fields

Script to Define Custom Fields:

```plaintext
status_message = "I'm a status message";
response.comments = "Comments field";
response.another_field = "Another field";
```

Request:

```
  -H "Accept: application/json" -H "Content-Type: application/json" --data
    '{"message":"some message", "uuid":"00007"}' -v
> POST /api/now/import/imp_notification HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> Accept: application/json
> Content-Type: application/json
</pre>

Response:

```
< HTTP/1.1 201 Created
< Location: http://https://<instance name>.service-now.com/api/now/import/imp_notification/49fa75019f23110041a496fcc67fcfe7
{
  "result": [
    {
      "sys_id": "01fa75019f23110041a496fcc67fcfe8",
      "status": "inserted",
      "record_link":
        "http://https://<instance name>.service-now.com/api/now/table/incident/01fa75019f23110041a496fcc67fcfe8",
      "display_value": "INC0010026",
      "display_name": "number",
      "table": "incident",
      "transform_map": "Notification"
    },
    {
      "comments": "Comments field",
      "status_message": "I'm a status message",
      "transform_map": "Problem Xform",
      "table": "problem",
      "display_name": "number",
      "display_name": "number",
      "display_value": "PRB0040027",
      "record_link":
      "status": "inserted",
      "sys_id": "cdfa75019f23110041a496fcc67fcfe8",
      "another_field": "Another field"
    }
  ],
  "staging_table": "imp_notification",
  "import_set": "ISET0010004"
}
```

Table 981: No update due to coalesce and no data change

Request:

```
-H "Accept: application/json" -H "Content-Type: application/json" --data '{"message":"some message", "uuid":"00007"}' -v
> POST /api/now/import/imp_notification HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> Accept: application/json
> Content-Type: application/json
>
Response:
< Location:
http://https://<instance name>.service-now.com/api/now/import/imp_notification/e82d75019f23110041a496fcc67f6c
{
 "result": [
 {
 "comments": "Comments field",
 "status_message": "I'm a status message;No field values changed",
 "transform_map": "Problem Xform",
 "table": "problem",
 "display_name": "number",
 "display_value": "PRB0040027",
 "record_link":
 http://https://<instance name>.service-now.com/api/now/table/problem/cdfa75019f23110041a496fcc67fcfe8",
 "status": "ignored",
 "sys_id": "cdafa75019f23110041a496fcc67fcfe8",
 "another_field": "Another field"
 },
 {
 "status_message": "No field values changed",
 "sys_id": "01fa75019f23110041a496fcc67fcfe8",
 "status": "ignored",
 "record_link":
 http://https://<instance name>.service-now.com/api/now/table/incident/01fa75019f23110041a496fcc67fcfe8",
 "display_value": "INC0040027",
 "display_name": "number",
 "table": "incident",
 "transform_map": "Notification"
 }
],
"staging_table": "imp_notification",
"import_set": "ISET0010004"
}

Table 982: Record updated

Request:
-H "Accept: application/json" -H "Content-Type: application/json" --data '{"message":"some message data changed", "uuid":"00007"}' -v
> POST /api/now/import/imp_notification HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> Accept: application/json
> Content-Type: application/json
>
Response:
< HTTP/1.1 201 Created
< Location:
http://https://<instance name>.service-now.com/api/now/import/imp_notification/dedd75019f23110041a496fcc67fcfe9

{  
"result": [  
{   
"comments": "Comments field",  
"status_message": "I'm a status message;No field values changed",  
"transform_map": "Problem Xform",  
"table": "problem",  
"display_name": "number",  
"display_value": "PRB0040027",  
"record_link":  
http://https://<instance name>.service-now.com/api/now/table/problem/cdfa75019f23110041a496fcc67fcfe8",  
"status": "ignored",  
"sys_id": "cdfa75019f23110041a496fcc67fcfe8",  
"another_field": "Another field"
},  
{   
"sys_id": "01fa75019f23110041a496fcc67fcfe8",  
"status": "updated",  
"record_link":  
http://https://<instance name>.service-now.com/api/now/table/incident/01fa75019f23110041a496fcc67fcfe8",  
"display_value": "INC0010026",  
"display_name": "number",  
"table": "incident",  
"transform_map": "Notification"
}
],  
"staging_table": "imp_notification",  
"import_set": "ISET0010004"
}

**GET Examples**

**Table 983: Getting the results again using the Location header link**

**Request:**
> GET /api/now/import/imp_notification/127375019f23110041a496fcc67fcfe3 HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> Accept: application/json
>
Response:

"result": [
  {
    "sys_id": "567375019f23110041a496fcc67fcfe3",
    "status": "inserted",
    "record_link": "http://https://<instance name>.service-now.com/api/now/table/incident/567375019f23110041a496fcc67fcfe3",
    "display_value": "INC0010023",
    "display_name": "number",
    "table": "incident",
    "transform_map": "Notification"
  }
],
"staging_table": "imp_notification",
"import_set": "ISET0010004"

**Attachment API**

The Attachment API allows you to upload and query file attachments.

You can upload or retrieve a single file with each request.

The Attachment API respects any system limitations on uploaded files, such as maximum file size and allowed attachment types. You can control these settings using the properties com.glide.attachment.max_size, 1024MB by default, and glide.attachment.extensions.

Attachment API - GET /now/attachment

This method gets the metadata for multiple attachments.

**URL format**

Versioned URL: api/now/v1/attachment

Default URL: api/now/attachment
### Supported request parameters

#### Table 984: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_query</td>
<td>An encoded query. Queries for the Attachment API are relative to the Attachments [sys_attachment] table.</td>
</tr>
<tr>
<td></td>
<td>For example: (sysparm_query=file_name=attachment.doc)</td>
</tr>
<tr>
<td></td>
<td>The encoded query provides support for order by. To sort responses based on certain fields, use the ORDERBY and ORDERBYDESC clauses in sysparm_query. For example, sysparm_query=ORDERBYfile_name^ORDERBYDESCtable_name orders the results in ascending order by name first, and then in descending order by table name.</td>
</tr>
<tr>
<td></td>
<td>If part of the query is invalid, such as by specifying an invalid field name, the instance ignores the invalid part. It then returns rows using only the valid portion of the query. You can control this behavior using the property glide.invalid_query.returns_no_rows. Set this property to true to return no rows on an invalid query.</td>
</tr>
<tr>
<td><strong>Note:</strong> This property controls the behavior of all queries across the instance, such as in lists, scripts (GlideRecord.query()), and web service APIs.</td>
<td></td>
</tr>
<tr>
<td>sysparm_limit</td>
<td>Limit to be applied on pagination. The default is 10000.</td>
</tr>
<tr>
<td></td>
<td>Unusually large sysparm_limit values can impact system performance.</td>
</tr>
<tr>
<td>sysparm_offset</td>
<td>A number of records to exclude from the query. Use this parameter when you need to get more records than specified in sysparm_limit. For example, if sysparm_limit is set to 500, but there are additional records you want to query, you can specify a sysparm_offset value of 500 to get the second set of records.</td>
</tr>
<tr>
<td>sysparm_read_replica_category</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sysparm_suppress_pagination_header</td>
<td>Set this value to <code>true</code> to remove the Link header from the response. The Link header enables you to request additional pages of data when the number of records matching your query exceeds the query limit.</td>
</tr>
</tbody>
</table>

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see [REST API headers](#).

**Table 985: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
</table>

**Table 986: Response headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>The content type of the response. For metadata requests, this is the content type of the metadata, not the content type of the attachment files.</td>
</tr>
<tr>
<td>Link</td>
<td>Links to download the attachments.</td>
</tr>
</tbody>
</table>

**Status codes**

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see [REST API response codes](#).

**Table 987: Status codes**

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the query ran successfully.</td>
</tr>
</tbody>
</table>

**Sample cURL request**

```bash
curl "https://instance.service-now.com/api/now/attachment?
sysparm_limit=1" \
--request GET \
--header "Accept:application/json" \
```
--user 'admin':'admin'

{
  "result": [
    {
      "table_sys_id": "5054b6f8c0a800060056addcf551ecf8",
      "size_bytes": "462",
      "download_link": "https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5/file",
      "sys_updated_on": "2009-05-21 04:12:21",
      "sys_id": "615ea769c0a80166001cf5f2367302f5",
      "image_height": "",
      "sys_created_on": "2009-05-21 04:12:21",
      "file_name": "blocks.swf",
      "sys_created_by": "glide.maint",
      "compressed": "true",
      "average_image_color": "",
      "sys_updated_by": "glide.maint",
      "sys_tags": "",
      "table_name": "content_block_programmatic",
      "image_width": "",
      "sys_mod_count": "0",
      "content_type": "application/x-shockwave-flash",
      "size_compressed": "485"
    }
  ]
}

Sample Python request

#Need to install requests package for python
#easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/attachment?
sysparm_limit=1'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:',response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
Attachment API - GET /now/attachment/{sys_id}
This method gets the metadata for the attachment file with a specific sys_id value.

**URL format**

Versioned URL: api/now/v1/attachment/<attachment record sys_id>
Default URL: api/now/attachment/<attachment record sys_id>

**Supported request parameters**

Table 988: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys_id</td>
<td>The sys_id of the attachment record you want to get metadata for.</td>
</tr>
</tbody>
</table>

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see [REST API headers](#).
Table 989: Request headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Table 990: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>The content type of the response. For metadata requests, this is the content type of the metadata, not the content type of the attachment files.</td>
</tr>
</tbody>
</table>

Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see `REST API response codes`.

Table 991: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the query ran successfully.</td>
</tr>
<tr>
<td>404</td>
<td>Indicates the specified attachment does not exist, or the current user cannot access it.</td>
</tr>
</tbody>
</table>

Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5" \
--request GET \
--header "Accept:application/json" \
--user 'admin':'admin'
```

```json
{
  "result": {
    "table_sys_id": "5054b6f8c0a800060056addcf551ecf8",
    "size_bytes": "462",
    "download_link": "https://instance.service.now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5/file",
    "sys_updated_on": "2009-05-21 04:12:21",
    "sys_id": "615ea769c0a80166001cf5f2367302f5",
    "image_height": "",
    "sys_created_on": "2009-05-21 04:12:21",
    "file_name": "blocks.swf",
    "sys_created_by": "glide.maint",
    "compressed": "true",
    "average_image_color": "",
    "sys_updated_by": "glide.maint",
  }
}
```
Sample Python request

```python
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:',response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <result>
    <table_sys_id>5054b6f8c0a800060056addcf551ecf8</table_sys_id>
    <size_bytes>462</size_bytes>
    <download_link>https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5/file</download_link>
    <sys_updated_on>2009-05-21 04:12:21</sys_updated_on>
    <sys_id>615ea769c0a80166001cf5f2367302f5</sys_id>
    <image_height />
    <sys_created_on>2009-05-21 04:12:21</sys_created_on>
    <file_name>blocks.swf</file_name>
    <sys_created_by>glide.maint</sys_created_by>
    <compressed>true</compressed>
    <average_image_color />
    <sys_updated_by>glide.maint</sys_updated_by>
    <sys_tags />
    <image_width />
    <table_name>content_block_programmatic</table_name>
```

© 2017 ServiceNow. All rights reserved. 3152
Attachment API - GET /now/attachment/{sys_id}/file
This method gets the binary file attachment with a specific sys_id value.

URL format

Versioned URL: api/now/v1/attachment/<attachment sys_id>/file
Default URL: api/now/attachment/<attachment sys_id>/file

Supported request parameters

Table 992: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys_id</td>
<td>The sys_id of the attachment record you want to get binary data from.</td>
</tr>
</tbody>
</table>

Headers

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

Table 993: Request headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Table 994: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-Attachment-Metadata</td>
<td>Metadata about the returned file, such as size, name, and file type.</td>
</tr>
</tbody>
</table>

Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see REST API response codes.
Table 995: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the query ran successfully.</td>
</tr>
</tbody>
</table>

Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5/file" 
--request GET 
--header "Accept:*/*" 
--user 'admin':'admin'```

Binary response not shown.

Sample Python request

```python
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5/file'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {'Content-Type':"application/xml","Accept":"*/*"}

# Do the HTTP request
response = requests.get(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:',response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```

Binary response not shown.

Attachment API - POST /now/attachment/file

This method uploads a binary file specified in the request body as an attachment.
URL format

Versioned URL: /api/now/v1/attachment/file
Default URL: /api/now/attachment/file

Supported request parameters

Table 996: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file_name (Required)</td>
<td>The name to give the attachment. This parameter is required to post an attachment.</td>
</tr>
<tr>
<td>table_name (Required)</td>
<td>The name of the table you want to attach the file to. This parameter is required to post an attachment.</td>
</tr>
<tr>
<td>table_sys_id (Required)</td>
<td>The sys_id of the record on the specified table that you want to attach the file to. This parameter is required to post an attachment.</td>
</tr>
<tr>
<td>encryption_context</td>
<td>The sys_id of an encryption context record. Specify this parameter to allow only users with the specified encryption context to access the attachment. If you do not specify this parameter, the attached file is not encrypted with any encryption context.</td>
</tr>
</tbody>
</table>

Headers

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

Table 997: Request headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>The content type of the file you want to attach. This header is mandatory to post file attachments.</td>
</tr>
</tbody>
</table>

Table 998: Response headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>The URL of the new attachment.</td>
</tr>
</tbody>
</table>
Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see REST API response codes.

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Indicates the query ran successfully.</td>
</tr>
<tr>
<td>400</td>
<td>Indicates that one or more mandatory parameters were missing from the request.</td>
</tr>
<tr>
<td>404</td>
<td>Indicates the record specified by the table_name and table_sys_id parameters does not exist or is not accessible by the current user.</td>
</tr>
</tbody>
</table>

Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/attachment/file?table_name=incident&table_sys_id=d71f7935c0a8016700802b64c67c11c6&file_name=Issue_screenshot" \
--request POST \
--header "Accept:application/json" \
--user 'admin':'admin' \
--header "Content-Type: image/jpeg" \
-F "uploadFile=$location of the file on file system"
```

```json
{
    "result": {
        "table_sys_id": "d71f7935c0a8016700802b64c67c11c6",
        "size_bytes": "36597",
        "download_link": "https://instance.service-now.com/api/now/attachment/6ea10fe64f411200adf9f8e18110c739/file",
        "sys_updated_on": "2016-01-22 15:14:07",
        "sys_id": "6ea10fe64f411200adf9f8e18110c739",
        "image_height": "",
        "sys_created_on": "2016-01-22 15:14:07",
        "file_name": "Issue_screenshot",
        "sys_created_by": "admin",
        "compressed": "true",
        "average_image_color": "",
        "sys_updated_by": "admin",
        "sys_tags": "",
        "table_name": "incident",
        "image_width": "",
        "sys_mod_count": "0",
        "content_type": "image/jpeg",
        "size_compressed": "25130"
    }
}
```
Sample Python request

```python
# Need to install requests package for python
# easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/attachment/file?
table_name=incident&table_sys_id=d71f7935c0a8016700802b64c67c11c6&file_name=Issue_screenshot.jpg'

# Specify the file to send. When specifying files to send make
# sure you specify the path to the file, in
# this example the file was located in the same directory as the
# python script being executed.
data = open('Issue_screenshot.jpg', 'rb').read()

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {'Content-Type': 'image/jpeg', 'Accept': 'application/json'}

# Do the HTTP request
response = requests.post(url, auth=(user, pwd), headers=headers, data=data)

# Check for HTTP codes other than 201
if response.status_code != 201:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:', response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)

{
"result": {
    "table_sys_id": "d71f7935c0a8016700802b64c67c11c6",
    "size_bytes": "36597",
    "download_link": "https://instance.service-now.com/api/now/attachment/6ea10fe64f411200adf9f8e18110c739/file",
    "sys_updated_on": "2016-01-22 15:14:07",
    "sys_id": "6ea10fe64f411200adf9f8e18110c739",
    "image_height": "",
    "sys_created_on": "2016-01-22 15:14:07",
    "file_name": "Issue_screenshot.jpg",
    "sys_created_by": "admin",
    "compressed": "true",
    "average_image_color": "",
    "sys_updated_by": "admin",
    "sys_tags": "",
    "table_name": "incident",
    "image_width": "",
    "sys_mod_count": "0",
    "content_type": "image/jpeg",
    "size_compressed": "25130"
}
```
Attachment API - POST /now/attachment/upload
This method uploads a multipart file attachment.

**URL format**

Versioned URL: /api/now/v1/attachment/upload
Default URL: /api/now/attachment/upload

**Supported request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>The multipart POST method does not accept any parameters. The table name and record sys_id values must be specified within the message body. See the POST multipart sample for an example of a multipart message.</td>
</tr>
</tbody>
</table>

**Important:** When using multipart POST, ensure the file content is contained in the final part of the message only. Earlier parts should contain only metadata such as table name and record sys_id.

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see REST API headers.

**Table 1001: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>The content type of the request. Set this value to multipart/form-data when using the multipart POST method.</td>
</tr>
</tbody>
</table>

**Table 1002: Response headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>The URL of the new attachment.</td>
</tr>
</tbody>
</table>
Status codes

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see REST API response codes.

Table 1003: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Indicates the query ran successfully.</td>
</tr>
</tbody>
</table>

POST multipart mandatory values

When sending a multipart POST request to upload a file attachment, include attachment data in the message body, not in the URL parameters. You must specify these values in the message body:

Table 1004: Mandatory values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table_name</td>
<td>The name of the table you want to attach the file to.</td>
</tr>
<tr>
<td>record_sys_id</td>
<td>The sys_id of the record on the specified table that you want to attach the file to.</td>
</tr>
<tr>
<td>Content-Type</td>
<td>The Content-Type of the file, included in the message body for multipart uploads.</td>
</tr>
</tbody>
</table>

**Note:** The Content-Type must be defined within the file portion of the POST message, not within the form data. See the sample POST multipart message for an example of a multipart message.

Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/attachment/upload" \
--request POST \
--header "Accept:application/json" \
--user 'admin':'admin'"\ 
--header "Content-Type:multipart/form-data" \
-F 'table_name=incident' -F 'table_sys_id=d71f7935c0a8016700802b64c67c11c6' -F 'encryption_context=undefined'-F 'uploadFile=\@ location of the file on file system'
```

```
{
  "result": {
    "table_sys_id": "d71f7935c0a8016700802b64c67c11c6",
    "size_bytes": "36597",
```
Sample Python request

```python
# This example uses the Python Requests Library and you will need to install requests package for python
# Documentation can be found at http://docs.python-requests.org/en/master/user/quickstart/
import requests
import pprint
import json

# Specify the Endpoint URL replacing instance with your ServiceNow Instance Name
url = 'https://instance.service-now.com/api/now/attachment/upload'

# Specify Parameters for File Being Uploaded, the table_name and table_sys_id should be replaced with values that make sense for your use case
payload = {'table_name': 'incident',
           'table_sys_id': '81f8915b6ba20028927416bf961971'}

# Specify Files To Send and Content Type. When specifying files to send make sure you specify the path to the file, in this example the file was located in the same directory as the python script being executed.
# it is important to specify the correct file type
files = {'file': ('issue_screenshot.JPG',
                  open('issue_screenshot.JPG', 'rb'), 'image/jpeg', {'Expires': '0'})}

# Eg. User name="admin", Password="admin" for this code sample. This will be sent across as basic authentication
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {'Accept': '*/*'}

# Send the HTTP request
response = requests.post(url, auth=(user, pwd), headers=headers, files=files, data=payload)
```
# Check for HTTP codes other than 201
if response.status_code != 201:
    print('Status:', response.status_code, 'Headers:',
    response.headers, 'Error Response:',response.json())
    exit()

# Print Response Details
print 'Response Status Code:', response.status_code
print ''
print('Response Payload:')
print json.dumps(response.json(), indent=4)

Response Status Code: 201
Response Payload:
{
    "result": {
        "sys_tags": "",
        "sys_updated_by": "admin",
        "content_type": "image/jpg",
        "sys_created_by": "admin",
        "file_name": "issue_screenshot.JPG",
        "sys_updated_on": "2017-01-05 10:47:09",
        "sys_created_on": "2017-01-05 10:47:09",
        "image_width": "",
        "image_height": "",
        "sys_mod_count": "0",
        "table_name": "incident",
        "sys_id": "96679f724f84320025e874828110c7bd",
        "download_link": "https://instance.service-now.com/api/
now/attachment/96679f724f84320025e874828110c7bd/file",
        "average_image_color": "",
        "size_bytes": "197484",
        "table_sys_id": "'81f8915bd6ba20028927416bf961971'",
        "size_compressed": "197005",
        "compressed": "true"
    }
}

Sample POST multipart message
This sample demonstrates a multipart POST message to the Attachment API.

Note that the file Content-Type value is defined within the file data.

POST /api/now/attachment/upload HTTP/1.1
Authorization: Basic YWRtaW46YWRtaW4=
Content-Type: multipart/form-data; boundary=WDCovLhyTgyWCrIdos2autXLLk8Qf8Pd
Host: <instance>.service-now.com
Connection: close
User-Agent: Paw/2.2.2 (Macintosh; OS X/10.10.3) GCDHTTPRequest
Content-Length: 425

--WDCovLhyTgyWCrIdos2autXLLk8Qf8Pd
Content-Disposition: form-data; name="table_name"
incident
--WDCovLhyTgyWCrIdos2autXLLk8Qf8Pd
Content-Disposition: form-data; name="table_sys_id"
Attachment API - DELETE /now/attachment/{sys_id}

This method deletes the attachment with a specific sys_id value.

**URL format**

**Versioned URL**: /api/now/v1/attachment/{sys_id}

**Default URL**: /api/now/attachment/{sys_id}

**Supported request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys_id</td>
<td>The sys_id value of the attachment to delete.</td>
</tr>
</tbody>
</table>

**Headers**

The following request and response headers apply to this HTTP action only, or apply to this action in a distinct way. For a list of general headers used in the REST API, see [REST API headers](#).

**Table 1006: Request headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1007: Response headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**Status codes**

The following status codes apply to this HTTP action. For a list of possible status codes used in the REST API, see [REST API response codes](#).
Table 1008: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>Indicates the request ran successfully.</td>
</tr>
</tbody>
</table>

Sample cURL request

```bash
curl "https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5" \
--request DELETE \
--header "Accept:application/json" \
--user 'admin':'admin'
```

Sample Python request

```python
#Need to install requests package for python
#easy_install requests
import requests

# Set the request parameters
url = 'https://instance.service-now.com/api/now/attachment/615ea769c0a80166001cf5f2367302f5'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/xml","Accept":"application/xml"}

# Do the HTTP request
response = requests.delete(url, auth=(user, pwd), headers=headers)

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers, 'Error Response:',response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```

Handling large attachments with the Attachment API

To upload or retrieve large attachments using the Attachment API you may need to configure the maximum transaction time.
Uploading or retrieving large attachments may take a long time depending on the attachment size and network speed. By default, if a transaction takes longer than 60 seconds, that transaction is cancelled and the error **Transaction cancelled: maximum execution time exceeded** appears.

If your integration routinely uploads or retrieves attachments greater than 50MB, increase the maximum transaction time to prevent transactions from being cancelled. An administrator can set the **Maximum duration** value of the REST Attachment API request timeout transaction quota.

**Performance Analytics API**

The Performance Analytics API allows you to query data about Performance Analytics scorecards.

The Performance Analytics API supports only the GET action. Performance Analytics queries never update records.

Performance Analytics API - GET /now/v1/pa/scorecards

This method retrieves Performance Analytics scorecard details.

**URL format**

- Versioned URL: /api/now/v1/pa/scorecards
- Default URL: /api/now/pa/scorecards

**Supported parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| sysparm_uuid      | Enter a colon-separated list of sys_id values to specify which indicators, breakdowns, and aggregates to query. The parameter follows this format:  
<indicator sys_id>[:breakdown sys_id]::<element sys_id>[:aggregate sys_id>  
The parameter must begin with the sys_id of an indicator record. Optionally, you can append the sys_id values of a breakdown and breakdown element to group the response based on the breakdown, and the sys_id of an aggregate to apply that aggregate. You can use a breakdown with an aggregate, or use only one.  

**Note:** If an indicator is configured to use a Default time series, all scorecards for that indicator use the selected aggregate.

See Performance Analytics API Examples for examples of fully-constructed sysparm_uuid values. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_breakdown</td>
<td>Enter the sys_id of a breakdown to return chart information organized as defined by the breakdown. For example, enter the sys_id of a priority breakdown to return separate task chart information for each priority value, such as Number of open incidents / Priority / 2 - High.</td>
</tr>
<tr>
<td>sysparm_include_scores</td>
<td>Set this parameter to true to return all scores for a scorecard. If a value is not specified, this parameter defaults to false and returns only the most recent score value.</td>
</tr>
<tr>
<td>sysparm_include_available_breakdowns</td>
<td>Set this parameter to true to return all available breakdowns for a scorecard. If a value is not specified, this parameter defaults to false and returns no breakdowns.</td>
</tr>
<tr>
<td>sysparm_include_available_aggregates</td>
<td>Set this parameter to true to return all available aggregates for a scorecard. If a value is not specified, this parameter defaults to false and returns no aggregates.</td>
</tr>
<tr>
<td>sysparm_display_value</td>
<td>Data retrieval operation for reference and choice fields. Based on this value, retrieves the display value and/or the actual value from the database.</td>
</tr>
<tr>
<td></td>
<td>• true returns display values for all fields.</td>
</tr>
<tr>
<td></td>
<td>• false returns actual values from the database. If a value is not specified, this parameter defaults to false.</td>
</tr>
<tr>
<td></td>
<td>• all returns both actual and display values.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> There is no preferred method for setting this parameter. However, specifying the display value may cause performance issues since it is not reading directly from the database and may include referencing other fields and records. For more information on display values and actual values, see Table API FAQs (KB0534905).</td>
</tr>
<tr>
<td>sysparm_exclude_reference_link</td>
<td>Set this parameter to true to hide additional information provided for reference fields, such as the URI to the reference resource.</td>
</tr>
<tr>
<td>sysparm_favorites</td>
<td>Set this parameter to true to return only scorecards that are favorites of the querying user.</td>
</tr>
<tr>
<td>sysparm_key</td>
<td>Set this parameter to true to return only scorecards for key indicators.</td>
</tr>
<tr>
<td>sysparm_target</td>
<td>Set this parameter to true to return only scorecards that have a target.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sysparm_display</td>
<td>Set this parameter to <code>true</code> to return only scorecards where the indicator <strong>Display</strong> field is selected. Set this parameter to <code>all</code> to return scorecards with any <strong>Display</strong> field value. This parameter is true by default.</td>
</tr>
<tr>
<td>sysparm_contains</td>
<td>Enter a comma-separated list of names or descriptions to return only scorecards with a matching value.</td>
</tr>
<tr>
<td>sysparm_tags</td>
<td>Enter a comma-separated list of sys_id values to return only scorecards with a matching sys_id.</td>
</tr>
<tr>
<td>sysparm_per_page</td>
<td>Enter the maximum number of scorecards each query can return. By default this value is 10, and the maximum is 100.</td>
</tr>
<tr>
<td>sysparm_page</td>
<td>Specify the page number. For example, when querying 20 scorecards with the default sysparm_per_page value (10), specify a sysparm_page value of 2 to retrieve scorecards 11-20.</td>
</tr>
<tr>
<td>sysparm_sortby</td>
<td>Specify the value to use when sorting results. Valid values for this parameter are value, change, changeperc, gap, gapperc, duedate, name, order, default, group, indicator_group, frequency, target, date, trend, bullet, and direction. By default, queries sort records by value.</td>
</tr>
<tr>
<td>sysparm_sortdir</td>
<td>Specify the sort direction, ascending or descending. By default, queries sort records in descending order. Use sysparm_sortdir=asc to sort in ascending order.</td>
</tr>
<tr>
<td>sysparm_elements_filter</td>
<td>Specify the sys_id of an elements filter to apply that filter to the returned data.</td>
</tr>
<tr>
<td>sysparm_breakdown_relation</td>
<td>Specify the sys_id of a breakdown relation to break down the returned data using that relation. You can view available breakdown relations by setting the sysparm_include_available_breakdowns parameter to <code>true</code>.</td>
</tr>
</tbody>
</table>

**Headers**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
Status codes

Table 1011: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the query ran successfully.</td>
</tr>
</tbody>
</table>

Performance Analytics API security
You must meet certain requirements to access the Performance Analytics REST API.

Access to tables via the REST API is restricted by BasicAuth. ACLs defined for tables are enforced to restrict access to data.

To make queries using the Performance Analytics API, you must also have the pa_viewer role.

Performance Analytics API examples
These examples demonstrate how to perform a REST query using cURL commands, and show the data returned for each command. Each example builds upon the last, with later examples using the data returned by earlier examples.

Return all main scorecards

You can request a list of all scorecards for indicators that have a Display value set to true.

Command:

```bash
curl -v -u "admin:admin" -H "Accept:application/json" "https://instance.service-now.com/api/now/v1/pa/scorecards"
```

Response:

```json
{
  "result" : [ 
    { 
      "change_formatted" : "", 
      "key" : true, 
      "value_unit" : ",", 
      "value_formatted" : "", 
      "period_title" : null, 
      "gapperc" : null, 
      "gap" : null, 
      "target" : null, 
      "period" : null, 
      "target_formatted" : "", 
      "favorite" : false, 
      "direction_label" : "Maximize", 
      "uuid" : "002d65c3d7131100b96d45a3ce6103e2", 
      "name" : "% of incidents resolved by first assigned group", 
      "value_color" : "#000000", 
      "frequency_label" : "Daily", 
      "change" : null, 
      "gap_formatted" : ",", 
      "gapperc_formatted" : ",", 
      "formula" : "(( [Number of resolved incidents by first assigned group] / [(Number of resolved incidents)] ) * 100", 
      "value" : null, 
      "unit" : {
```
"display_value": ":\%",
"link": "https://instance.service-now.com/api/now/v1/table/pa_units/f9c365e2d7320100ba986f14ce6103b8",
"value": "f9c365e2d7320100ba986f14ce6103b8"
},
"changeperc_formatted": ":\%",
"direction": 3,
"frequency": 10,
"precision": 2,
"changeperc": null,
"indicator": {
  "display_value": "% of incidents resolved by first assigned group",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/002d65c3d7131100b96d45a3ce6103e2",
  "value": "002d65c3d7131100b96d45a3ce6103e2"
},
"description": "Percentage of incidents resolved by first assigned group.
"},
{,
  "change_formatted": ":\%",
  "key": true,
  "value_unit": ":\%",
  "value_formatted": ":\%",
  "period_title": null,
  "gapperc": null,
  "gap": null,
  "target": null,
  "period": null,
  "target_formatted": ":\%",
  "favorite": false,
  "direction_label": "Minimize",
  "uuid": "4660f602d7130100b96d45a3ce610383",
  "name": "% of new critical incidents",
  "value_color": ":#000000",
  "frequency_label": "Daily",
  "change": null,
  "gap_formatted": ":\%",
  "gapperc_formatted": ":\%",
  "formula": "(\[\[Number of new incidents / Priority / 1 - Critical\]\] / 
[[Number of new incidents]] ) * 100",
  "value": null,
  "unit": {
    "display_value": ":\%",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_units/f9c365e2d7320100ba986f14ce6103b8",
    "value": "f9c365e2d7320100ba986f14ce6103b8"
  },
  "changeperc_formatted": ":\%",
  "direction": 2,
  "frequency": 10,
  "precision": 2,
  "changeperc": null,
  "indicator": {
    "display_value": "% of new critical incidents",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/4660f602d7130100b96d45a3ce610383",
    "value": "4660f602d7130100b96d45a3ce610383"
  },
  "description": "Number of new critical incidents as a percentage of number of new incidents.
"},
{,
  "change_formatted": ":\%",
  "key": true,
  "value_unit": ":\%",
  "value_formatted": ":\%",
  "period_title": null,
  "gapperc": null,
  "gap": null,
  "target": null,
  "period": null,
  "target_formatted": ":\%",
  "favorite": false,
  "direction_label": "Minimize",
  "uuid": "4660f602d7130100b96d45a3ce610383",
  "name": "% of new critical incidents",
  "value_color": ":#000000",
  "frequency_label": "Daily",
  "change": null,
  "gap_formatted": ":\%",
  "gapperc_formatted": ":\%",
  "formula": "(\[\[Number of new incidents / Priority / 1 - Critical\]\] / 
[[Number of new incidents]] ) * 100",
  "value": null,
  "unit": {
    "display_value": ":\%",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_units/f9c365e2d7320100ba986f14ce6103b8",
    "value": "f9c365e2d7320100ba986f14ce6103b8"
  },
  "changeperc_formatted": ":\%",
  "direction": 2,
  "frequency": 10,
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Period Title</th>
<th>Favorite</th>
<th>Direction Label</th>
<th>UUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of open incidents not updated in last 30 days</td>
<td><img src="https://instance.service-now.com/api/now/v1/table/pa_units/f9c365e2d7320100ba986f14ce6103b8" alt="Math expression" /></td>
<td></td>
<td></td>
<td>Minimize</td>
<td>f0f07202d7130100b96d45a3ce610383</td>
</tr>
<tr>
<td>% of open incidents not updated in last 5 days</td>
<td><img src="https://instance.service-now.com/api/now/v1/table/pa_units/fd51f602d7320100ba986f14ce6103b8" alt="Math expression" /></td>
<td></td>
<td></td>
<td>Minimize</td>
<td>fd51f602d7130100b96d45a3ce610385</td>
</tr>
</tbody>
</table>

**Description:** Number of open incidents not updated in last 30 days as a percentage of number of open incidents.
"change_formatted" : "",
"key" : true,
"value_unit" : "",
"value_formatted" : "",
"period_title" : null,
"gapperc" : null,
"gap" : null,
"target" : null,
"period" : null,
"target_formatted" : "",
"favorite" : false,
"direction_label" : "Minimize",
"uuid" : "6fbb7202d7130100b96d45a3ce610360",
"name" : "Average resolution time of resolved incidents",
"value_color" : "#000000",
"frequency_label" : "Daily",
"change" : null,
"gap_formatted" : "",
"gapperc_formatted" : "",
"formula" : "\[
\frac{\text{Summed duration of resolved incidents}}{\text{Number of resolved incidents}} / 24
\]",
"value" : null,
"unit" : {
   "display_value" : "Days",
   "link" : "https://instance.service-now.com/api/now/v1/table/pa_units/94d365e2d7320100ba986f14ce6103be",
   "value" : "94d365e2d7320100ba986f14ce6103be"
},
"changeperc_formatted" : "",
"direction" : 2,
"frequency" : 10,
"precision" : 2,
"changeperc" : null,
"indicator" : {
   "display_value" : "Average resolution time of resolved incidents",
   "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/6fbb7202d7130100b96d45a3ce610360",
   "value" : "6fbb7202d7130100b96d45a3ce610360"
},
"description" : "Average resolution time of resolved incidents"
},
{
   "change_formatted" : "",
   "key" : true,
   "value_unit" : "",
   "value_formatted" : "",
   "period_title" : null,
   "gapperc" : null,
   "gap" : null,
   "target" : null,
   "period" : null,
   "target_formatted" : "",
   "favorite" : false,
   "direction_label" : "Minimize",
   "uuid" : "d0b0f602d7130100b96d45a3ce6103b0",
   "name" : "Incident backlog growth",
   "value_color" : "#000000",
   "frequency_label" : "Daily",
   "change" : null,
   "gap_formatted" : "",
   "gapperc_formatted" : "",
   "formula" : "\[
\text{Number of new incidents} - \text{Number of resolved incidents}\\n\]",
   "value" : null,
   "unit" : {
      "display_value" : "\text{Number of new incidents} - \text{Number of resolved incidents}\\n\]",
      "link" : "https://instance.service-now.com/api/now/v1/table/pa_units/d0b0f602d7130100b96d45a3ce6103b0",
      "value" : "d0b0f602d7130100b96d45a3ce6103b0"
   },
   "changeperc_formatted" : "",
   "direction" : 2,
   "frequency" : 10,
   "precision" : 2,
   "changeperc" : null,
   "indicator" : {
      "display_value" : "Incident backlog growth",
      "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/d0b0f602d7130100b96d45a3ce6103b0",
      "value" : "d0b0f602d7130100b96d45a3ce6103b0"
   },
   "description" : "Incident backlog growth"}
"unit": {  
  "display_value": "#",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
  "value": "17b365e2d7320100ba986f14ce6103ad"
},
"changeperc_formatted": "",
"direction": 2,
"frequency": 10,
"precision": 0,
"changeperc": null,
"indicator": {
  "display_value": "Number of new incidents",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/31efe602d7130100b96d45a3ce610300",
  "value": "31efe602d7130100b96d45a3ce610300"
},
"description": "Number of incidents based on registration date."}
"value_formatted" : "",
"period_title" : null,
"gapperc" : null,
"gap" : null,
"target" : null,
"period" : null,
"targetFormatted" : "",
"favorite" : false,
"direction_label" : "Minimize",
"uuid" : "fb007202d7130100b96d45a3ce6103b4",
"name" : "Number of open incidents",
"value_color" : "#000000",
"frequency_label" : "Daily",
"change" : null,
"gap_formatted" : "",
"gapperc_formatted" : "",
"value" : null,
"unit" : {
  "display_value" : "#",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
  "value" : "17b365e2d7320100ba986f14ce6103ad"
},
"changeperc_formatted" : "",
"direction" : 2,
"frequency" : 10,
"precision" : 0,
"changeperc" : null,
"indicator" : {
  "display_value" : "Number of open incidents",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
  "value" : "fb007202d7130100b96d45a3ce6103b4"
},
"description" : "Number of incidents open based on resolved date is empty."
},
{
  "change_formatted" : "",
  "key" : false,
  "value_unit" : "",
  "value_formatted" : "",
  "period_title" : null,
  "gapperc" : null,
  "gap" : null,
  "target" : null,
  "period" : null,
  "targetFormatted" : "",
  "favorite" : false,
  "direction_label" : "Minimize",
  "uuid" : "44944f12bf130100b96d4b8f96d45a3ce6103b4",
  "name" : "Number of open incidents not updated in last 30 days",
  "value_color" : "#000000",
  "frequency_label" : "Daily",
  "change" : null,
  "gap_formatted" : "",
  "gapperc_formatted" : "",
  "value" : null,
  "unit" : {
    "display_value" : "#",
    "link" : "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
    "value" : "17b365e2d7320100ba986f14ce6103ad"
  }
}
"changeperc_formatted" : ",
"direction" : 2,
"frequency" : 10,
"precision" : 0,
"changeperc" : null,
"indicator" : {
    "display_value" : "Number of open incidents not updated in last 30 days",
    "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/44944f12bf130100b96dac808c0739a7",
    "value" : "44944f12bf130100b96dac808c0739a7"
},
"description" : "Number of open incidents not updated in last 30 days based on updated date."
}
]

Return the number of open incidents scorecard

You can query scorecards for a particular indicator by providing the sysparm_uuid parameter with an indicator sys_id value.

Command:

```
curl -v -u "admin:admin" -H "Accept:application/json" 
"https:///instance.service-now.com/api/now/v1/pa/scorecards? 
sysparm_uuid=fb007202d7130100b96d45a3ce6103b4"
```

Response:

```
{
  "result" : [
    {
      "change_formattted" : "",
      "key" : false,
      "value_unit" : "",
      "value_formattted" : "",
      "period_title" : null,
      "gapperc" : null,
      "gap" : null,
      "target" : null,
      "period" : null,
      "target_formattted" : "",
      "favorite" : false,
      "direction_label" : "Minimize",
      "uuid" : "fb007202d7130100b96d45a3ce6103b4",
      "name" : "Number of open incidents",
      "value_color" : 
"#000000",
      "frequency_label" : "Daily",
      "change" : null,
      "gap_formattted" : "",
      "gapperc_formattted" : "",
      "value" : null,
      "unit" : {
        "display_value" : ",
        "link" : "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
        "value" : "17b365e2d7320100ba986f14ce6103ad"
      },
      "changeperc_formattted" : ",
```
"direction" : 2,
"frequency" : 10,
"precision" : 0,
"changeperc" : null,
"indicator" : {
   "display_value" : "Number of open incidents",
   "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
   "value" : "fb007202d7130100b96d45a3ce6103b4"
},
"description" : "Number of incidents open based on resolved date is empty."
}
[

Return the scorecard with all breakdowns and aggregates

You can query a list of available breakdowns and aggregates for an indicator by setting the sysparm_include_available_breakdowns and sysparm_include_available_aggregates parameters to true.

Command:

curl -v -u "admin:admin" -H "Accept:application/json" "https://instance.service-now.com/api/now/v1/pa/scorecards?sysparm_uuid=fb007202d7130100b96d45a3ce6103b4&sysparm_include_available_breakdowns=true&sysparm_include_available_aggregates=true"

Response:

{
   "result" : [
      {
         "key" : false,
         "change_formatted" : 
         "aggregates" : [
            {
               "display_value" : "7d running SUM",
               "link" : "https://instance.service-now.com/api/now/v1/table/pa_aggregates/89ea4c11d7001100ba986f14ce6103dc",
               "value" : "89ea4c11d7001100ba986f14ce6103dc"
            },
            {
               "display_value" : "28d running SUM",
               "link" : "https://instance.service-now.com/api/now/v1/table/pa_aggregates/4dfa4c11d7001100ba986f14ce6103e2",
               "value" : "4dfa4c11d7001100ba986f14ce6103e2"
            },
            {
               "display_value" : "30d running SUM",
               "link" : "https://instance.service-now.com/api/now/v1/table/pa_aggregates/3e409011d7001100ba986f14ce610319",
               "value" : "3e409011d7001100ba986f14ce610319"
            },
            {
               "display_value" : "7d running AVG",
               "link" : "https://instance.service-now.com/api/now/v1/table/pa_aggregates/9ef05051d7001100ba986f14ce610372",
               "value" : "9ef05051d7001100ba986f14ce610372"
            },
            {
               "display_value" : "28d running AVG",
               "link" : "https://instance.service-now.com/api/now/v1/table/pa_aggregates/aef05051d7001100ba986f14ce610372",
               "value" : "aef05051d7001100ba986f14ce610372"
            }
         ]
      }
   ]
}
"link": "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
"value": "17b365e2d7320100ba986f14ce6103ad",
"breakdowns": [
    {
        "display_value": "Priority",
        "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
        "value": "0df47e02d7130100b96d45a3ce610399"
    },
    {
        "display_value": "Category",
        "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/1f918835d7231100b96d45a3ce6103fe",
        "value": "1f918835d7231100b96d45a3ce6103fe"
    },
    {
        "display_value": "Assignment Group",
        "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed",
        "value": "baec0752bf130100b96dac808c0739ed"
    },
    {
        "display_value": "State",
        "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/f0647e02d7130100b96d45a3ce61030b",
        "value": "f0647e02d7130100b96d45a3ce61030b"
    },
    {
        "display_value": "Age",
        "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/65947e02d7130100b96d45a3ce61033a",
        "value": "65947e02d7130100b96d45a3ce61033a"
    }
],
"changeperc_formatted": "",
"frequency": 10,
"precision": 0,
"direction": 2,
"indicator": {
    "display_value": "Number of open incidents",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
    "value": "fb007202d7130100b96d45a3ce6103b4"
},
"description": "Number of incidents open based on resolved date is empty."
]
}

Return the scorecard with breakdown relations

You can obtain the sys_id values for all breakdown relations associated with the scorecard using the sysparm_include_available_breakdowns parameter.
**Command:**

curl -v -u "admin:admin" -H "Accept:application/json" 
"https://<instance>.service-now.com/api/now/v1/pa/scorecards?sysparm_uuid=fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:287ee6fe9e198100ada7950d0b1b73&sysparm_include_available_breakdowns=true"

**Response:**

```json
{
    "result": [
        {
            "value_formatted": "37",
            "indicator": {
                "display_value": "Number of open incidents",
                "link": "https://<instance>.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
                "value": "fb007202d7130100b96d45a3ce6103b4"
            },
            "gapperc": null,
            "change": 9.0,
            "value_color": "#455464",
            "direction": 2,
            "target_formatted": "",
            "frequency": 10,
            "changeperc_formatted": "32.1%",
            "direction_label": "Minimize",
            "period_title": "Jul 22",
            "description": "Number of incidents open based on resolved date is empty."
        },
        {
            "name": "Number of open incidents / Assignment Group / Database",
            "value": 37.0,
            "key": false,
            "element": {
                "display_value": "Database",
                "link": "https://<instance>.service-now.com/api/now/v1/table/sys_user_group/287ee6fe9e198100ada7950d0b1b73",
                "value": "287ee6fe9e198100ada7950d0b1b73"
            },
            "precision": 0,
            "breakdowns": [
                {
                    "display_value": "Priority",
                    "link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
                    "value": "0df47e02d7130100b96d45a3ce610399"
                },
                {
                    "display_value": "Category",
                    "link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/1f918835d7231100b96d45a3ce6103fe",
                    "value": "1f918835d7231100b96d45a3ce6103fe"
                },
                {
                    "display_value": "State",
                    "link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/f0647e02d7130100b96d45a3ce61030b",
                    "value": "f0647e02d7130100b96d45a3ce61030b"
                },
                {
                    "display_value": "Age",
                    "link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/65947e02d7130100b96d45a3ce61033a"
                }
            ]
        }
    ]
}
```
Return the scorecard broken down using a breakdown relation

You can break down the returned data by passing a breakdown relation sys_id in the sysparm_breakdown_relation parameter.
Command:

curl -v -u "admin:admin" -H "Accept:application/json" 
"https://<instance>.service-now.com/api/now/v1/pa/scorecards?
sysparm_uuid=fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:db53580b0b550100b86d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:db53580b0b550100b86d45a3ce6103b4

Response:

{
  "result": [
    {
      "value_formatted": "37",
      "indicator": {
        "display_value": "Number of open incidents",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
        "value": "fb007202d7130100b96d45a3ce6103b4"
      },
      "gapperc": null,
      "change": 9.0,
      "value_color": "#455464",
      "direction": 2,
      "target_formatted": "",
      "frequency": 10,
      "changeperc_formatted": "32.1%",
      "direction_label": "Minimize",
      "period_title": "Jul 22",
      "description": "Number of incidents open based on resolved date is empty.",
      "name": "Number of open incidents / Assignment Group / Database",
      "value": 37.0,
      "key": false,
      "gap_formated": "",
      "element": {
        "display_value": "Database",
        "link": "https://<instance>.service-now.com/api/now/v1/table/sys_user_group/287ee6fe9fe198100ada7950d0b1b73",
        "value": "287ee6fe9fe198100ada7950d0b1b73"
      },
      "precision": 0,
      "breakdown": {
        "display_value": "Assignment Group",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed",
        "value": "baec0752bf130100b96dac808c0739ed"
      },
      "period": "Jul 22",
      "favorite": false,
      "change_formatted": "9",
      "unit": {
        "display_value": "#",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
        "value": "17b365e2d7320100ba986f14ce6103ad"
      },
      "frequency_label": "Daily",
      "target": null,
      "changeperc": 0.32142857142857145,
      "uuid": "fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:287ee6fe9fe198100ada7950d0b1b73:baec0752bf130100b96dac808c0739ed:17b365e2d7320100ba986f14ce6103ad";
    }
  ]
}
Return the scorecard broken down by location

The Performance Analytics API returns geolocation data when available.

Command:

```
curl -v -u "admin:admin" -H "Accept:application/json" 
  "https://<instance>.service-now.com/api/now/v1/pa/scorecards? 
sysparm_uuid=fb007202d7130100b96d45a3ce6103b4&sysparm_breakdown=656d5662eb23310065deac6aa206fee7"
```

Response:

```
{
  "result": [
    {
      "element": {
        "display_value": "San Diego",
        "link": "https://<instance>.service-now.com/api/now/v1/table/cmnd_location/108752c8c611227501d4ab0e392ba97f",
        "value": "108752c8c611227501d4ab0e392ba97f",
        "longitude": -117.15726,
        "latitude": 32.71533
      },
      ...
    },
    {
      "element": {
        "display_value": "Florida",
        "link": "https://<instance>.service-now.com/api/now/v1/table/cmnd_location/8e3e85f037d02000044e0bfc8bcbe5d14",
        "value": "8e3e85f037d02000044e0bfc8bcbe5d14",
        "longitude": -95.71289,
        "latitude": 37.09024
      },
      ...
    }]
}
```

Return the scorecard with a filter

You can apply a filter to the scorecard data using the `sysparm_elements_filter` parameter with the sys_id of a Performance Analytics element filter record.

Command:

```
curl -v -u "admin:admin" -H "Accept:application/json" 
  "https://<instance>.service-now.com/api/now/v1/pa/scorecards? 
sysparm_uuid=fb007202d7130100b96d45a3ce6103b4&sysparm_breakdown=baec0752bf130100b96dac80379f2e6&sysparm_elements_filter=7b9eb563eb11020065deac6aa206fe11"
```
Response:

{
  "result": [
    {
      "value_formatted": "37",
      "indicator": {
        "display_value": "Number of open incidents",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
        "value": "fb007202d7130100b96d45a3ce6103b4"
      },
      "gapperc": null,
      "change": 9.0,
      "value_color": "#455464",
      "direction": 2,
      "target_formatted": "",
      "frequency": 10,
      "changeperc_formatted": "32.1%",
      "direction_label": "Minimize",
      "period_title": "Jul 22",
      "description": "Number of incidents open based on resolved date is empty.",
      "name": "Number of open incidents / Assignment Group / Database",
      "value": 37.0,
      "key": false,
      "gap_formatted": "",
      "element": {
        "display_value": "Database",
        "link": "https://<instance>.service-now.com/api/now/v1/table/sys_user_group/287ee6fe9fe198100ada7950d0b1b73",
        "value": "287ee6fe9fe198100ada7950d0b1b73"
      },
      "precision": 0,
      "breakdown": {
        "display_value": "Assignment Group",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/287ee6fe9fe198100ada7950d0b1b73",
        "value": "287ee6fe9fe198100ada7950d0b1b73"
      },
      "period": "Jul 22",
      "favorite": false,
      "change_formatted": "9",
      "unit": {
        "display_value": ",",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
        "value": "17b365e2d7320100ba986f14ce6103ad"
      },
      "frequency_label": "Daily",
      "target": null,
      "changeperc": 0.32142857142857145,
      "uuid": "fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:287ee6fe9fe198100ada7950d0b1b73",
      "gapperc_formatted": "",
      "value_unit": "37",
      "gap": null
    },
    {
      "value_formatted": "20",
      "indicator": {
        "display_value": "Number of open incidents",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
        "value": "fb007202d7130100b96d45a3ce6103b4"
      },
      "gapperc": null,
      "change": 11.0,
      "value_color": "#4b92a3",
      "direction": 1,
      "target_formatted": "",
      "frequency": 10,
      "changeperc_formatted": "44.4%",
      "direction_label": "Maximize",
      "period_title": "Jul 22",
      "description": "Number of incidents open based on resolved date is empty.",
      "name": "Number of open incidents / Assignment Group / Database",
      "value": 20.0,
      "key": false,
      "gap_formatted": "",
      "element": {
        "display_value": "Database",
        "link": "https://<instance>.service-now.com/api/now/v1/table/sys_user_group/287ee6fe9fe198100ada7950d0b1b73",
        "value": "287ee6fe9fe198100ada7950d0b1b73"
      },
      "precision": 0,
      "breakdown": {
        "display_value": "Assignment Group",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/287ee6fe9fe198100ada7950d0b1b73",
        "value": "287ee6fe9fe198100ada7950d0b1b73"
      },
      "period": "Jul 22",
      "favorite": false,
      "change_formatted": "11",
      "unit": {
        "display_value": ",",
        "link": "https://<instance>.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
        "value": "17b365e2d7320100ba986f14ce6103ad"
      },
      "frequency_label": "Daily",
      "target": null,
      "changeperc": 0.4438016038016038,
      "uuid": "fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:287ee6fe9fe198100ada7950d0b1b73",
      "gapperc_formatted": "",
      "value_unit": "37",
      "gap": null
    }
  ]
}
Number of incidents open based on resolved date is empty.

**name**: Number of open incidents / Assignment Group / Database Atlanta

- **value**: 20.0
- **key**: false
- **precision**: 0
- **breakdown**: { display_value: "Assignment Group", link: "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed", value: "baec0752bf130100b96dac808c0739ed" }
- **period**: "Jul 22"
- **frequency_label**: "Daily"
- **target**: null
- **uuid**: "fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:db53580b0a0a6501aa37c294a2ba6b"
- **gapperc**: null
- **change**: 4.0
- **value_color**: "#455464"
- **direction**: 2
- **target_formatted**: "\n"
"frequency": 10,
"changeperc_formatted": "35.7%",
"direction_label": "Minimize",
"period_title": "Jul 22",
"description": "Number of incidents open based on resolved date is empty.",
"name": "Number of open incidents / Assignment Group / NY DB",
"value": 19.0,
"key": false,
"gap_formatted": "",
"element": {
"display_value": "NY DB",
"link": "https://<instance>.service-now.com/api/now/v1/table/sys_user_group/5f74727dc0a8010e01efe33a251993f9",
"value": "5f74727dc0a8010e01efe33a251993f9"
},
"precision": 0,
"breakdown": {
"display_value": "Assignment Group",
"link": "https://<instance>.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed",
"value": "baec0752bf130100b96dac808c0739ed"
},
"period": "Jul 22",
"favorite": false,
"change_formatted": "5",
"unit": {
"display_value": "#",
"link": "https://<instance>.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
"value": "17b365e2d7320100ba986f14ce6103ad"
},
"frequency_label": "Daily",
"target": null,
"changeperc": 0.35714285714285715,
"uuid": "fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:5f74727dc0a8010e01efe33a251993f9",
"gapperc_formatted": "",
"value_unit": "19",
"gap": null
},
{ "value_formatted": "10",
"indicator": {
"display_value": "Number of open incidents",
"link": "https://<instance>.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
"value": "fb007202d7130100b96d45a3ce6103b4"
},
"gapperc": null,
"change": 0.0,
"value_color": "#455464",
"direction": 2,
"target_formatted": "",
"frequency": 10,
"changeperc_formatted": "0.0%",
"direction_label": "Minimize",
"period_title": "Jul 22",
"description": "Number of incidents open based on resolved date is empty.",
"name": "Number of open incidents / Assignment Group / Database San Diego",
"value": 10.0,
"key": false,
Return the scorecard with an aggregate

You can apply the 7d running SUM aggregate to the scorecard using the sysparm_uuid value with the sys_id of the aggregate.

Command:

curl -v -u "admin:admin" -H "Accept:application/json" 
"https://instance.service-now.com/api/now/v1/pa/scorecards?
sysparm_uuid=fb007202d7130100b96d45a3ce6103b4:baec0752bf130100b96dac808c0739ed:db53a9290a0a650091abebccf833c6"

Response:

{
  "result" : [
    {
      "key" : false,
      "change_formatted" : "",
      "value_unit" : "",
      "value_formatted" : "",
      "period_title" : null,
      "aggregate" : {
        "display_value" : "7d running SUM",
        "link" : "https://instance.service-now.com/api/now/v1/table/pa_aggregates/89ea4c11d7001100ba986f14ce6103dc",
        "value" : "89ea4c11d7001100ba986f14ce6103dc"
      }
    }
  ]
}
Return the scorecard with priority breakdown

You can request broken down scorecard data by passing the sysparm_breakdown parameter. This example shows the Number of open incidents scorecard broken down by priority.

In this example, the [PA Incident] Daily Data Collection job must run at least once to populate the data.

Command:

```
curl -v -u "admin:admin" -H "Accept:application/json" 
  "https://instance.service-now.com/api/now/v1/pa/scorecards?sysparm_uuid=fb007202d7130100b96d45a3ce6103b4&sysparm_breakdown=0df47e02d7130100b96d45a3ce610399"
```

Response:

```
{
  "result" : [
    {
      "key" : false,
      "change_formatted" : "0",
      "changeperc" : 0,
      "changeperc_formatted" : "",
      "direction" : 2,
      "frequency" : 10,
      "precision" : 0,
      "changeperc" : null,
      "indicator" : {
        "display_value" : "Number of open incidents",
        "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
        "value" : "fb007202d7130100b96d45a3ce6103b4"
      },
      "description" : "Number of incidents open based on resolved date is empty."
    }
  ]
}
```
"value_unit": "15",
"value_formatted": "15",
"period_title": "Mar 23",
"gapperc": null,
"gap": null,
"target": null,
"period": "Mar 23",
"target_formatted": ",",
"favorite": false,
"direction_label": "Minimize",
"uuid": "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:dce1db9cc803310026c149f3d065511",
"name": "Number of open incidents / Priority / 1 - Critical",
"value_color": "#000000",
"frequency_label": "Daily",
"element": {
  "display_value": "1 - Critical",
  "link": "https://instance.service-now.com/api/now/v1/table/sys_choice/dce1db9cc803310026c149f3d065511",
  "value": "dce1db9cc803310026c149f3d065511"
},
"change": 0,
"gap_formatted": ",",
"gapperc_formatted": ",",
"value": 15,
"unit": {
  "display_value": "#",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_units/17b36e2d7320100ba986f14ce6103ad",
  "value": "17b36e2d7320100ba986f14ce6103ad"
},
"breakdown": {
  "display_value": "Priority",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
  "value": "0df47e02d7130100b96d45a3ce610399"
},
"changeperc_formatted": "0.0%",
"frequency": 10,
"precision": 0,
"direction": 2,
"indicator": {
  "display_value": "Number of open incidents",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
  "value": "fb007202d7130100b96d45a3ce6103b4"
},
"description": "Number of incidents open based on resolved date is empty."
},

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuid</td>
<td>fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:50e1db9cc803310026c1c49f3d065512</td>
</tr>
<tr>
<td>name</td>
<td>Number of open incidents / Priority / 3 - Moderate</td>
</tr>
<tr>
<td>value_color</td>
<td>#000000</td>
</tr>
<tr>
<td>frequency_label</td>
<td>Daily</td>
</tr>
</tbody>
</table>
| element             | { display_value: "3 - Moderate", link: "https://instance.service-now.com/api/now/v1/table/sys_choice/50e1db9cc803310026c1c49f3d065512", value: "50e1db9cc803310026c1c49f3d065512" }, change: 0, gap_formatted: "", gapperc_formatted: "", value: 5, unit: { display_value: ", link: "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba98f14ce6103ad", value: "17b365e2d7320100ba98f14ce6103ad" }, breakdown: { display_value: "Priority", link: "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399", value: "0df47e02d7130100b96d45a3ce610399" }, changeperc_formatted: "0.0%", frequency: 10, precision: 0, direction: 2, indicator: { display_value: "Number of open incidents", link: "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4", value: "fb007202d7130100b96d45a3ce6103b4" }, description: "Number of incidents open based on resolved date is empty." }, 
| key                 | false                                                                 |
| change_formatted    | "0"                                                                   |
| changeperc          | 0                                                                     |
| value_unit          | "4"                                                                   |
| value_formatted     | "4"                                                                   |
| period_title        | "Mar 23"                                                              |
| gap                 | null                                                                  |
| target              | null                                                                  |
| period              | "Mar 23"                                                              |
| target_formatted    | ""                                                                    |
| favorite            | false                                                                 |
| direction_label     | "Minimize"                                                            |
| uuid                | fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:10e1db9cc803310026c1c49f3d065512 |
| name                | Number of open incidents / Priority / 2 - High                         |
| value_color         | #000000                                                              |
| frequency_label     | Daily                                                                 |
| element             | { display_value: "2 - High", link: "https://instance.service-now.com/api/now/v1/table/sys_choice/10e1db9cc803310026c1c49f3d065512", value: "10e1db9cc803310026c1c49f3d065512" } |
Return the scorecard with priority breakdown and available breakdowns

You can request broken down scorecard data by passing the sysparm_breakdown parameter, and a list of available breakdowns by passing the sysparm_include_available_breakdowns parameter. Passing both parameters in the same request allows you to query both sets of data using a single request.

In this example, the [PA Incident] Daily Data Collection job must run at least once to populate the data.

Command:
```
curl -v -u "admin:admin" -H "Accept:application/json" "https://instance.service-now.com/api/now/v1/pa/scorecards?sysparm_uuid=fb007202d7130100b96d45a3ce6103b4&sysparm_breakdown=0df47e02d7130100b96d45a3ce610399&sysparm_include_available_breakdowns=true"
```

Response:
```
{
    "result" : [ {
        "key" : false,
        "direction" : 2,
        "change_formatted" : "0",
        "changeperc" : 0,
        "value_formatted" : "15",
        "period_title" : "Mar 23",
        "gapperc" : null,
        "value_unit" : "15",
        "target" : null,
        "period" : "Mar 23",
        "target_formatted" : ",",
        "favorite" : false,
        "gap" : null,
        "direction_label" : "Minimize",
        "uuid" : "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:dce1db9cc803310026c1c49f3d065511",
        "name" : "Number of open incidents / Priority / 1 - Critical",
        "value_color" : "#000000",
        "frequency_label" : "Daily",
        "element" : { 
            "display_value" : "1 - Critical",
            "link" : "https://instance.service-now.com/api/now/v1/table/sys_choice/dce1db9cc803310026c1c49f3d065511",
            "value" : "dce1db9cc803310026c1c49f3d065511"
        },
        "change" : 0,
        "gap_formatted" : ",",
        "gapperc_formatted" : ",",
        "value" : 15,
    }
}
```
"unit": {
  "display_value": "#",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
  "value": "17b365e2d7320100ba986f14ce6103ad"
},
"breakdown": {
  "display_value": "Priority",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
  "value": "0df47e02d7130100b96d45a3ce610399"
},
"breakdowns": [
  {
    "display_value": "Category",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/1f918835d7231100b96d45a3ce6103fe",
    "value": "1f918835d7231100b96d45a3ce6103fe"
  },
  {
    "display_value": "Assignment Group",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed",
    "value": "baec0752bf130100b96dac808c0739ed"
  },
  {
    "display_value": "State",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/f0647e02d7130100b96d45a3ce61030b",
    "value": "f0647e02d7130100b96d45a3ce61030b"
  },
  {
    "display_value": "Age",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/65947e02d7130100b96d45a3ce61033a",
    "value": "65947e02d7130100b96d45a3ce61033a"
  }
],
"changeperc_formatted": "0.0%",
"precision": 0,
"frequency": 10,
"indicator": {
  "display_value": "Number of open incidents",
  "link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
  "value": "fb007202d7130100b96d45a3ce6103b4"
},
"description": "Number of incidents open based on resolved date is empty."}
"direction_label" : "Minimize",
"uuid" :
"fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:50e1db9cc803310026c1c49f3d065512",
"name" : "Number of open incidents / Priority / 3 - Moderate",
"value_color" : "#000000",
"frequency_label" : "Daily",
"element" : {
  "display_value" : "3 - Moderate",
  "link" : "https://instance.service-now.com/api/now/v1/table/sys_choice/50e1db9cc803310026c1c49f3d065512",
  "value" : "50e1db9cc803310026c1c49f3d065512"
},
"change" : 0,
"gap_formatted" : "",
"gapperc_formatted" : "",
"value" : 5,
"unit" : {
  "display_value" : "#",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
  "value" : "17b365e2d7320100ba986f14ce6103ad"
},
"breakdown" : {
  "display_value" : "Priority",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
  "value" : "0df47e02d7130100b96d45a3ce610399"
},
"breakdowns" : [
  {
    "display_value" : "Category",
    "link" : "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/1f918835d7231100b96d45a3ce6103fe",
    "value" : "1f918835d7231100b96d45a3ce6103fe"
  },
  {
    "display_value" : "Assignment Group",
    "link" : "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed",
    "value" : "baec0752bf130100b96dac808c0739ed"
  },
  {
    "display_value" : "State",
    "link" : "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/f0647e02d7130100b96d45a3ce61030b",
    "value" : "f0647e02d7130100b96d45a3ce61030b"
  },
  {
    "display_value" : "Age",
    "link" : "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/65947e02d7130100b96d45a3ce61033a",
    "value" : "65947e02d7130100b96d45a3ce61033a"
}
],
"changeperc_formatted" : "0.0%",
"precision" : 0,
"frequency" : 10,
"indicator" : {
  "display_value" : "Number of open incidents",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
  "value" : "fb007202d7130100b96d45a3ce6103b4"
}
"description": "Number of incidents open based on resolved date is empty."
},
{
  "key": false,
  "direction": 2,
  "change_formatted": "0",
  "changeperc": 0,
  "value_formatted": "4",
  "period_title": "Mar 23",
  "gapperc": null,
  "value_unit": "4",
  "target": null,
  "period": "Mar 23",
  "target_formatted": "",
  "favorite": false,
  "gap": null,
  "direction_label": "Minimize",
  "uuid": "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:10e1db9cc803310026c1c49f3d065512",
  "name": "Number of open incidents / Priority / 2 - High",
  "value_color": "#000000",
  "frequency_label": "Daily",
  "element": {
    "display_value": "2 - High",
    "link": "https://instance.service-now.com/api/now/v1/table/sys_choice/10e1db9cc803310026c1c49f3d065512",
    "value": "10e1db9cc803310026c1c49f3d065512"
  },
  "change": 0,
  "gap_formatted": "",
  "gapperc_formatted": "",
  "value": 4,
  "unit": {
    "display_value": "#",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
    "value": "17b365e2d7320100ba986f14ce6103ad"
  },
  "breakdown": {
    "display_value": "Priority",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
    "value": "0df47e02d7130100b96d45a3ce610399"
  },
  "breakdowns": [
    {
      "display_value": "Category",
      "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/1f918835d7231100b96d45a3ce6103fe",
      "value": "1f918835d7231100b96d45a3ce6103fe"
    },
    {
      "display_value": "Assignment Group",
      "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed",
      "value": "baec0752bf130100b96dac808c0739ed"
    },
    {
      "display_value": "State",
      "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/f0647e02d7130100b96d45a3ce61030b",
      "value": "f0647e02d7130100b96d45a3ce61030b"
    },
  ]
Geneva ServiceNow ServiceNow Platform

{
    "display_value": "Age",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/65947e02d7130100b96d45a3ce61033a",
    "value": "65947e02d7130100b96d45a3ce61033a"
}

{
    "display_value": "Number of open incidents",
    "link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
    "value": "fb007202d7130100b96d45a3ce6103b4"
}

"description": "Number of incidents open based on resolved date is empty."

{
    "key": false,
    "direction": 2,
    "change_formatted": "0",
    "changeperc": 0,
    "value_formatted": "3",
    "period_title": "Mar 23",
    "gapperc": null,
    "value_unit": "3",
    "target": null,
    "period": "Mar 23",
    "target_formatted": "",
    "favorite": false,
    "gap": null,
    "direction_label": "Minimize",
    "uuid": "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:d0e1db9cc803310026c1c49f3d065512",
    "name": "Number of open incidents / Priority / 5 - Planning",
    "value_color": "#000000",
    "frequency_label": "Daily",
    "element": {
        "display_value": "5 - Planning",
        "link": "https://instance.service-now.com/api/now/v1/table/sys_choice/d0e1db9cc803310026c1c49f3d065512",
        "value": "d0e1db9cc803310026c1c49f3d065512"
    },
    "change": 0,
    "gap_formatted": "",
    "gapperc_formatted": "",
    "value": 3,
    "unit": {
        "display_value": "#",
        "link": "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
        "value": "17b365e2d7320100ba986f14ce6103ad"
    },
    "breakdown": {
        "display_value": "Priority",
        "link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
        "value": "0df47e02d7130100b96d45a3ce610399"
    },
    "breakdowns": [
        {
            "display_value": "Category",
            "value": "Category"
        }
    ]
}
{"link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/1f918835d7231100b96d45a3ce6103fe",
"value": "1f918835d7231100b96d45a3ce6103fe"
},
{
"display_value": "Assignment Group",
"link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/baec0752bf130100b96dac808c0739ed",
"value": "baec0752bf130100b96dac808c0739ed"
},
{
"display_value": "State",
"link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/f0647e02d7130100b96d45a3ce61030b",
"value": "f0647e02d7130100b96d45a3ce61030b"
},
{
"display_value": "Age",
"link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/65947e02d7130100b96d45a3ce61033a",
"value": "65947e02d7130100b96d45a3ce61033a"
}
],
"changeperc_formatted": "0.0%",
"precision": 0,
"frequency": 10,
"indicator": {
"display_value": "Number of open incidents",
"link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
"value": "fb007202d7130100b96d45a3ce6103b4"
},
"description": "Number of incidents open based on resolved date is empty."
},
{
"key": false,
"direction": 2,
"change_formatted": "0",
"changeperc": 0,
"value_formatted": "1",
"period_title": "Mar 23",
"gapperc": null,
"value_unit": "1",
"target": null,
"period": "Mar 23",
"target_formatted": "",
"favorite": false,
"gap": null,
"direction_label": "Minimize",
"uuid": "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:90e1db9cc803310026c1c49f3d065512",
"name": "Number of open incidents / Priority / 4 - Low",
"value_color": 
"frequency_label": "Daily",
"element": {
"display_value": "4 - Low",
"link": "https://instance.service-now.com/api/now/v1/table/sys_choice/90e1db9cc803310026c1c49f3d065512",
"value": "90e1db9cc803310026c1c49f3d065512"
},
"change": 0,
"gap_formatted": "",
"gapperc_formatted": ""}
Return the scorecard for priority 1 incidents

You can apply a breakdown by appending the breakdown and breakdown element sys_id values to the sysparm_uuid parameter. In this example, the data is broken down to show priority 1 incidents.
Command:

```
curl -v -u "admin:admin" -H "Accept:application/json"
  "https://instance.service-now.com/api/now/v1/pa/scorecards?
sysparm_uuid=fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:e5900140200331007665978299a805f3"
```

Response:

```
{
  "result": [
    {
      "value_formatted": "",
      "indicator": {
        "display_value": "Number of open incidents",
        "link": "http://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
        "value": "fb007202d7130100b96d45a3ce6103b4"
      },
      "gapperc": null,
      "change": null,
      "value_color": "#000000",
      "direction": 2,
      "target_formatted": "",
      "frequency": 10,
      "changeperc_formatted": "",
      "direction_label": "Minimize",
      "period_title": null,
      "description": "Number of incidents open based on resolved date is empty.",
      "name": "Number of open incidents / Priority / 1 - Critical",
      "value": null,
      "key": false,
      "gap_formatted": "",
      "element": {
        "display_value": "1 - Critical",
        "link": "http://instance.service-now.com/api/now/v1/table/sys_choice/e5900140200331007665978299a805f3",
        "value": "e5900140200331007665978299a805f3"
      },
      "precision": 0,
      "breakdown": {
        "display_value": "Priority",
        "link": "http://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
        "value": "0df47e02d7130100b96d45a3ce610399"
      },
      "period": null,
      "favorite": false,
      "change_formatted": "",
      "unit": {
        "display_value": "#",
        "link": "http://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
        "value": "17b365e2d7320100ba986f14ce6103ad"
      },
      "frequency_label": "Daily",
      "target": null,
      "changeperc": null,
      "uuid": "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:e5900140200331007665978299a805f3",
      "gapperc_formatted": "",
      "value_unit": "",
      "gap": null
    }
  ]
}
```
**Return the scorecard for priority 1 database incidents**

You can apply multiple breakdowns by appending multiple breakdown sys_ids to the sysparm_uuid parameter. In this example, the data is broken down by priority to show priority 1 incidents, and by category to show database incidents.

**Command:**

```bash
curl -v -u "admin:admin" -H "Accept:application/json" "https://instance.service-now.com/api/now/v1/pa/scorecards?sysparm_uuid=fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:e5900140200331007665978299a805f3:1f918835d7231100b96d45a3ce6103fe:9e418d40200331007665978299a805c1"
```

**Response:**

```json
{
  "result": [
    {
      "value_formatted": 
      "indicator": {
        "display_value": "Number of open incidents",
        "link": "http://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
        "value": "fb007202d7130100b96d45a3ce6103b4"
      },
      "gapperc": null,
      "change": null,
      "value_color": 
      "direction": 2,
      "target_formatted": 
      "frequency": 10,
      "changeperc_formatted": 
      "direction_label": "Minimize",
      "period_title": null,
      "description": "Number of incidents open based on resolved date is empty.",
      "name": "Number of open incidents / Priority / 1 - Critical / Category / Database",
      "value": null,
      "key": false,
      "gap_formatted": 
      "element": {
        "display_value": "1 - Critical",
        "link": "http://instance.service-now.com/api/now/v1/table/sys_choice/e5900140200331007665978299a805f3",
        "value": "e5900140200331007665978299a805f3"
      },
      "precision": 0,
      "element_level2": {
        "display_value": "Database",
        "link": "http://instance.service-now.com/api/now/v1/table/sys_choice/9e418d40200331007665978299a805c1",
        "value": "9e418d40200331007665978299a805c1"
      },
      "breakdown": {
        "display_value": "Priority",
        "link": "http://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399"
      }
    }
  ]
}
```
Return the scorecard for priority 1 database incidents with all scores

You can request a list of individual scores by setting the sysparm_include_scores parameter to true.

Command:

```
curl -v -u "admin:admin" -H "Accept:application/json" "https://instance.service-now.com/api/now/v1/pa/scorecards?sysparm_uuid=fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:e5900140200331007665978299a805f3:1f918835d7231100b96d45a3ce6103fe:9e418d40200331007665978299a805c1&sysparm_include_scores=true"
```

Response:

```
{
  "result": [
    {
      "key": false,
      "change_formatted": "0",
      "changeperc": 0,
      "value_unit": "15",
      "value_formatted": "15",
      "period_title": "Mar 23",
      "gapperc": null,
      "gap": null,
      "target": null,
      "period": "Mar 23",
      "target_formatted": "",
      "favorite": false,
      "scores": [
        {
          "end_at": "2015-03-23",
          "period": "Mar 23",
          "start_at": "2015-03-23",
```

© 2017 ServiceNow. All rights reserved.
"value_formatted": "15",
"value": 15
},
{
"end_at": "2015-03-22",
"period": "Mar 22",
"start_at": "2015-03-22",
"value_formatted": "15",
"value": 15
},
{
"end_at": "2015-03-21",
"period": "Mar 21",
"start_at": "2015-03-21",
"value_formatted": "22",
"value": 22
},
{
"end_at": "2015-03-20",
"period": "Mar 20",
"start_at": "2015-03-20",
"value_formatted": "22",
"value": 22
},
{
"end_at": "2015-03-19",
"period": "Mar 19",
"start_at": "2015-03-19",
"value_formatted": "22",
"value": 22
},
{
"end_at": "2015-03-18",
"period": "Mar 18",
"start_at": "2015-03-18",
"value_formatted": "22",
"value": 22
},
{
"end_at": "2015-03-17",
"period": "Mar 17",
"start_at": "2015-03-17",
"value_formatted": "22",
"value": 22
},
{
"end_at": "2015-03-16",
"period": "Mar 16",
"start_at": "2015-03-16",
"value_formatted": "22",
"value": 22
}
],
"direction_label": "Minimize",
"uuid": "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:dce1db9cc803310026c1c49f3d065511",
"name": "Number of open incidents / Priority / 1 - Critical",
"value_color": "#000000",
"frequency_label": "Daily",
"element": {
"display_value": "1 - Critical",
"link": "https://instance.service-now.com/api/now/v1/table/sys_choice/dce1db9cc803310026c1c49f3d065511",
"value": "dce1db9cc803310026c1c49f3d065511"
}
{ "key": false, "change_formatted": "0", "changeperc": 0, "value_unit": "4", "value_formatted": "4", "period_title": "Mar 23", "gapperc": null, "gap": null, "target": null, "period": "Mar 23", "target_formatted": "", "favorite": false, "scores": [ { "end_at": "2015-03-23", "period": "Mar 23", "start_at": "2015-03-23", "value_formatted": "4", "value": 4 } ], { "end_at": "2015-03-22", "period": "Mar 22", "start_at": "2015-03-22", "value_formatted": "4", "value": 4 } ], { "end_at": "2015-03-21", "period": "Mar 21", "start_at": "2015-03-21", "value_formatted": "4", "value": 4 } }


},

{  
  "end_at" : "2015-03-20",
  "period" : "Mar 20",
  "start_at" : "2015-03-20",
  "value_formatted" : "4",
  "value" : 4
},

{  
  "end_at" : "2015-03-19",
  "period" : "Mar 19",
  "start_at" : "2015-03-19",
  "value_formatted" : "4",
  "value" : 4
},

{  
  "end_at" : "2015-03-18",
  "period" : "Mar 18",
  "start_at" : "2015-03-18",
  "value_formatted" : "4",
  "value" : 4
},

{  
  "end_at" : "2015-03-17",
  "period" : "Mar 17",
  "start_at" : "2015-03-17",
  "value_formatted" : "4",
  "value" : 4
}
]
"direction_label" : "Minimize",
"uuid" : "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:10e1db9cc803310026c1c49f3d065512",
"name" : "Number of open incidents / Priority / 2 - High",
"value_color" : ":#000000",
"frequency_label" : "Daily",
"element" : {
  "display_value" : "2 - High",
  "link" : "https://instance.service-now.com/api/now/v1/table/sys_choice/10e1db9cc803310026c1c49f3d065512",
  "value" : "10e1db9cc803310026c1c49f3d065512"
},
"change" : 0,
"gap_formatted" : "",
"gapperc_formatted" : "",
"value" : 4,
"unit" : {
  "display_value" : "#",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
  "value" : "17b365e2d7320100ba986f14ce6103ad"
},
"breakdown" : {
  "display_value" : "Priority",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
  "value" : "0df47e02d7130100b96d45a3ce610399"
}
"changeperc_formatted" : "0.0%",
"frequency" : 10,
"precision" : 0,
"direction" : 2,
"indicator" : {
  "display_value" : "Number of open incidents",
  "link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
  "value" : "fb007202d7130100b96d45a3ce6103b4"
},
"description" : "Number of incidents open based on resolved date is empty."
},
{
  "key" : false,
  "change_formatted" : "0",
  "changeperc" : 0,
  "value_unit" : "5",
  "value_formatted" : "5",
  "period_title" : "Mar 23",
  "gapperc" : null,
  "gap" : null,
  "target" : null,
  "period" : "Mar 23",
  "target_formatted" : "",
  "favorite" : false,
  "scores" : [
    {
      "end_at" : "2015-03-23",
      "period" : "Mar 23",
      "start_at" : "2015-03-23",
      "value_formatted" : "5",
      "value" : 5
    },
    {
      "end_at" : "2015-03-22",
      "period" : "Mar 22",
      "start_at" : "2015-03-22",
      "value_formatted" : "5",
      "value" : 5
    },
    {
      "end_at" : "2015-03-21",
      "period" : "Mar 21",
      "start_at" : "2015-03-21",
      "value_formatted" : "6",
      "value" : 6
    },
    {
      "end_at" : "2015-03-20",
      "period" : "Mar 20",
      "start_at" : "2015-03-20",
      "value_formatted" : "6",
      "value" : 6
    },
    {
      "end_at" : "2015-03-19",
      "period" : "Mar 19",
      "start_at" : "2015-03-19",
      "value_formatted" : "6",
      "value" : 6
    }
  ]
"end_at" : "2015-03-18",
"period" : "Mar 18",
"start_at" : "2015-03-18",
"value_formatted" : "6",
"value" : 6
},
{
"end_at" : "2015-03-17",
"period" : "Mar 17",
"start_at" : "2015-03-17",
"value_formatted" : "6",
"value" : 6
},
{
"end_at" : "2015-03-16",
"period" : "Mar 16",
"start_at" : "2015-03-16",
"value_formatted" : "6",
"value" : 6
}
],
"direction_label" : "Minimize",
"uuid" : "fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:50e1db9cc803310026c1c49f3d065512",
"name" : "Number of open incidents / Priority / 3 - Moderate",
"value_color" : ":#000000",
"frequency_label" : "Daily",
"element" : {
"display_value" : "3 - Moderate",
"link" : "https://instance.service-now.com/api/now/v1/table/sys_choice/50e1db9cc803310026c1c49f3d065512",
"value" : "50e1db9cc803310026c1c49f3d065512"
},
"change" : 0,
"gap_formatted" : ":",
"gapperc_formatted" : "",
"value" : 5,
"unit" : {
"display_value" : ":#",
"link" : "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
"value" : "17b365e2d7320100ba986f14ce6103ad"
},
"breakdown" : {
"display_value" : "Priority",
"link" : "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
"value" : "0df47e02d7130100b96d45a3ce610399"
},
"changeperc_formatted" : "0.0%",
"frequency" : 10,
"precision" : 0,
"direction" : 2,
"indicator" : {
"display_value" : "Number of open incidents",
"link" : "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
"value" : "fb007202d7130100b96d45a3ce6103b4"
},
"description" : "Number of incidents open based on resolved date is empty."
},
{
"key" : false,
"change_formatted": "0",
"changeperc": 0,
"value_unit": "1",
"value_formatted": "1",
"period_title": "Mar 23",
"gapperc": null,
"gap": null,
"target": null,
"period": "Mar 23",
"target_formatted": "",
"favorite": false,
"scores": [
    
    
    
    
    
    
    
    
]
"start_at": "2015-03-16",
"value_formatted": "3",
"value": 3
}
],
"direction_label": "Minimize",
"uuid": 
"fb007202d7130100b96d45a3ce6103b4:0df47e02d7130100b96d45a3ce610399:90e1db9cc803310026c1c49f3d065512",
"name": "Number of open incidents / Priority / 4 - Low",
"value_color": "#000000",
"frequency_label": "Daily",
"element": {
"display_value": "4 - Low",
"link": "https://instance.service-now.com/api/now/v1/table/sys_choice/90e1db9cc803310026c1c49f3d065512",
"value": "90e1db9cc803310026c1c49f3d065512"
},
"change": 0,
"gap_formatted": "",
"gapperc_formatted": "",
"value": 1,
"unit": {
"display_value": "#",
"link": "https://instance.service-now.com/api/now/v1/table/pa_units/17b365e2d7320100ba986f14ce6103ad",
"value": "17b365e2d7320100ba986f14ce6103ad"
},
"breakdown": {
"display_value": "Priority",
"link": "https://instance.service-now.com/api/now/v1/table/pa_breakdowns/0df47e02d7130100b96d45a3ce610399",
"value": "0df47e02d7130100b96d45a3ce610399"
},
"changeperc_formatted": "0.0%",
"frequency": 10,
"precision": 0,
"direction": 2,
"indicator": {
"display_value": "Number of open incidents",
"link": "https://instance.service-now.com/api/now/v1/table/pa_indicators/fb007202d7130100b96d45a3ce6103b4",
"value": "fb007202d7130100b96d45a3ce6103b4"
},
"description": "Number of incidents open based on resolved date is empty."
},
{  "key": false,
  "change_formatted": "0",
  "changeperc": 0,
  "value_unit": "3",
  "value_formatted": "3",
  "period_title": "Mar 23",
  "gapperc": null,
  "gap": null,
  "target": null,
  "period": "Mar 23",
  "target_formatted": "",
  "favorite": false,
  "scores": [
    {
      "end_at": "2015-03-23",
      "period": "Mar 23",
      "start_at": "2015-03-23",
    }
  ]
User Role Inheritance API

The User Role Inheritance API allows you to see the roles that a specific user inherited.

A user’s role can be directly granted, inherited from other roles, or inherited from groups.

You must have the user_admin role to access this API.

User Role Inheritance API - GET /global/user_role_inheritance

This method retrieves the roles the user has and determines what was inherited.

URL format

Versioned URL: api/global/v1/user_role_inheritance

Default URL: api/global/user_role_inheritance

Supported parameters

Table 1012: Supported parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_sysid</td>
<td>The Sys ID of the user.</td>
</tr>
</tbody>
</table>
Status codes

Table 1013: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the query ran successfully.</td>
</tr>
</tbody>
</table>

Sample request

The response body shows you this information about the user's role:

- **user_name**: the username for the Sys ID you queried.
- **from_role**: the roles that are either granted directly or inherited from other roles.
- **from_group**: the roles that are inherited from groups that have roles.

The following is an example using modified demo data:

```
{
    "result": {
        "user_name": "abel.tuter",
        "from_group": [
            "[Azure Operators]/sn_azure.operator",
            "[Azure Operators]/sn_azure.operator/sn_azure.user"
        ],
        "from_role": [
            "/snc_internal",
            "/cloud_user",
            "/cloud_user/pa_viewer",
            "/cloud_user/sn_azure.user",
            "/cloud_user/aws_user"
        ]
    }
}
```

**IdentifyReconcile API**

The IdentifyReconcile API uses the Identification and Reconciliation framework to minimize creation of duplicate CIs and to reconcile CI attributes by only accepting information from authorized sources when updating the CMDB.

**IdentifyReconcile API - POST**

Insert or update configuration items in the CMDB based on identification and reconciliation rules. Use this API instead of updating the CMDB directly.

**URL format**

- Versioned URL: `/api/now/v1/identifyreconcile`
- Default URL: `/api/now/identifyreconcile`
Supported request parameters

Table 1014: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_data_source</td>
<td>Identifies the source of the CI information. This must be one of the choice values defined for the discovery_source field of the cmdb_ci table.</td>
</tr>
</tbody>
</table>

Headers

Table 1015: Request headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
<td>Specify application/json</td>
</tr>
<tr>
<td>Content-Type</td>
<td>Specify application/json</td>
</tr>
</tbody>
</table>

Table 1016: Response headers

Input payload

A JSON formatted string of configuration items to be added or updated. Each input string is in the format 'items: [{}], relations:[{}]', where each item within the items and relations lists contains name-value pairs.

The possible name-value pairs within the items list are:

- className - the sys_class_name of the CI to be created or updated.
- values:{ } - the field information for the CI as name-value pairs, where the name is the field name.
- lookup:[{}] - a list of records with each item having name-value pairs like the items list.

The possible name-value pairs within the relations list are:

- parent - index of the parent item in the dependency relation
- child - index of the child item in the dependency relation
- type - the relationship type. This is one of the name field values from the cmdb_rel_type table.

Sample input payload

The sample input is a JSON string that contains a list of configuration items and a list of relationships that exist between these configuration items.

```json
{"items": [ {"className":"cmdb_ci_web_server",
  "values": {"name":"apache linux den 200",
  "running_process_command": "xyz"},
```
"running_process_parameters": "abc",
"tcp_port": "3452"},
{"className": "cmdb_ci_linux_server",
 "values": {"name": "linux100", "ram": "2048"}},

{"relations": [{"parent": "0", "child": "1", "type": "Runs on::Runs"}]}

**Response payload**

A JSON formatted string that is a list of results for the configuration items in the input string. Each result string is in the format 'items: [{},], relations: [{}]', where each item within the items and relations lists contains name-value pairs.

The possible name-value pairs within the items list are:

- **className** - the sys_class_name for the CI to be updated or created.
- **operation**, which is one of INSERT, UPDATE, UPDATE_WITH_UPGRADE, UPDATE_WITH_DOWNGRADE, UPDATE_WITH_SWITCH, DELETE, NO_CHANGE
- **sysId** - the sys_id of the CI that was updated or created.
- **relatedSysIds** - a list of sys_id values of CIs used during lookup based identification.
- **identifierEntrySysId** - sys_id of identifier entry used during matching.
- **errors** - a list of errors in the format of (error, message string), where error can be ABANDONED, INVALID_INPUT_DATA, IDENTIFICATION_RULE_MISSING, IDENTIFICATION_RULE_FOR_LOOKUP_MISSING, NO_LOOKUP_RULES_FORDEPENDENT_CI, NO_CLASS_NAME_FOR_INDEPENDENT_CI, MISSING_DEPENDENCY, MULTIPLE_DEPENDENCIES, MULTIPLE_DUPLICATE_RECORDS, RELATION_CHAIN_ENDS_ATQUALIFIER, QUALIFICATION_LOOP, TYPE_CONFLICT_INQUALIFICATION, MULTI_MATCH, REQUIREDATTRIBUTE_EMPTY, RECLASSIFICATION_NOT_ALLOWED
- **duplicateIndices** - a list of indexes of items that are duplicates of the current item.
- **identificationAttempts** - a list of attempts in the format of (attributes, identifierName, attemptResult, searchOnTable) where
  - **attributes** - the attributes used during identification
  - **identifierName** - the CI identifier to which this identifier belongs
  - **attemptResult** - one of SKIPPED, NO_MATCH, MATCHED, MULTI_MATCH
  - **searchOnTable** - the table searched during the identification process.

The possible name-value pairs within the relations list are:

- **className** - the relationship CI's class name and is always cmdb_rel_ci
- **operation** - one of INSERT, UPDATE, NO_CHANGE
- **sysId** - the sys_id of the relationship CI inserted or updated

**Sample responses**

Line feeds have been added for presentation.

```json
{"items": [{"className": "cmdb_ci_web_server",
 "operation": "UPDATE",
 "sysId": "5f8af237c0a8010e01a932999468b83a",
 "identifierEntrySysId": "8985a23ec3f00200d8d4bea192d3ae08"},
```

© 2017 ServiceNow. All rights reserved. 3213
Status codes

Table 1017: Status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates the request completed successfully.</td>
</tr>
</tbody>
</table>

REST API version

REST API URIs may specify a version number, such as /api/now/v1/table/(tablename) or no version number, such as /api/now/table/(tableName).

URIs with a version number provide a consistent interface that is guaranteed to maintain backwards compatibility in future ServiceNow releases. For example, if the default response format for a particular HTTP method changes in a release, versioned URIs continue to use the previous format.

By specifying a version number in your URIs, you can ensure that future updates to the REST API do not negatively impact your integration. URIs that do not specify a version number use the most recent REST API behavior available with your instance version, which may change when you upgrade.

REST API security

By default, the REST API uses basic authentication or OAuth to enforce access controls to web resources. ACLs defined for tables are enforced to restrict access to data.

The user ID that is used for authentication is subject to access control in the same way as an interactive user. Each request requires the proper authentication information. Ensure each request includes an Authorization header with the credentials you want to use. There is no support for inbound mutual authentication.
To allow access to tables without any authentication and authorization, add the table name to sys_public.list. ACLs defined on tables are still enforced, and it is the customer’s responsibility to deactivate ACLs on tables.

REST supports cookies for binding to the existing session.

For Aggregate API requests, you must have read access for all records in the table you query. If an ACL prevents the requesting user from accessing any record in the table, the request returns a 403 Forbidden error.

REST API roles
Certain roles are required for inbound REST operations.

Table 1018: Inbound REST API roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>web_service_admin</td>
<td>Can access the System Web Services application menu and the REST modules.</td>
</tr>
</tbody>
</table>

**Note:** To perform database operations using REST, you must have any roles needed to access the target tables.

Attachment API role requirements
The user record used to authenticate Attachment API requests must have certain roles.

To create attachments, the user record used to authenticate the HTTP request with ServiceNow must have any roles required to create Attachment [sys_attachment] records, and also have any roles required to read and write records on the target table, such as the itil role to add attachments to incident records.

By default there is no single role allowing a user to add attachments. You can create a role to explicitly allow adding attachments, then assign this role to the user account being used to make the request.

REST API table access
All tables, including base system tables, global tables, and scoped tables are accessible via web services by default.

You must fulfill any other web service security requirements, such as basic authentication and ACLs to access tables via web services.

You can control direct web service access to tables using the **Allow access to this table via web services** check box on the table application access settings. This check box must be selected to allow web service interaction with the table.

**Note:** The application access fields controlling CRUD operations, such as Can read or Can create do not apply to web service requests.

REST API CORS support
The REST API supports cross-origin resource sharing (CORS) security.

CORS support allows you to define which domains can access each REST API. By defining a CORS rule, you can whitelist a domain to allow cross-origin requests from that domain. Cross-origin requests cannot be made from domains without a CORS rule.

**Note:** CORS support applies only to REST APIs, including scripted REST web services. Other web service APIs, such as the SOAP API, do not support CORS.
You can configure CORS to allow access to only certain APIs, HTTP methods, and headers from other domains. For example, you can limit requests to the Table API from a specific domain to allow only GET operations.

To view the CORS rules defined on your instance, navigate to **System Web Services > CORS Rules**. You can disable CORS support for an instance by setting the property glide.rest.cors.enabled to false. When false, no CORS evaluation is performed on incoming REST requests. This property is true by default.

Define a CORS rule
You can define a CORS rule to control which domains can access specific REST API endpoints.

Role required: web_service_admin

1. Navigate to **System Web Services > CORS Rules**.
2. Click **New**.
3. Populate the form.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST API</td>
<td>Select the REST API this CORS rule applies to, such as the Table API.</td>
</tr>
<tr>
<td>Domain</td>
<td>Enter the domain that this CORS rule applies to. This CORS rule is evaluated against requests from the specified domain.</td>
</tr>
<tr>
<td></td>
<td>You can specify a domain pattern or an IP address. When using a domain pattern you can specify a single wildcard to match incoming origin headers.</td>
</tr>
<tr>
<td>HTTP Methods</td>
<td>Select the HTTP methods allowed. Only the selected methods can be called from the specified domain.</td>
</tr>
<tr>
<td>HTTP Headers</td>
<td>Enter a comma-separated list of HTTP headers to send in the response. Specified headers are added to the Access-Control-Expose-Headers header.</td>
</tr>
<tr>
<td>Max age</td>
<td>Enter the number of seconds to cache the client session. After an initial CORS request, further requests from the same client within the specified time do not require a preflight message. If you do not specify a value, the default value of 0 indicates that all requests require a preflight message.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

CORS domain requirements
When you define a CORS rule, the value you enter in the Domain field must meet certain requirements. Each CORS rule supports a single wildcard to match incoming Origin headers.
Requirements

The value you enter in the Domain field on the CORS Rule form must meet the following requirements.

- Begins with HTTP:// or HTTPS://.
- Is a domain pattern or IP address.
- Ends with alphanumeric characters preceded by a period, such as .com.
- Includes at most a single wildcard character immediately following the scheme and hierarchical portion of the domain pattern.

Wildcard

You can use a single wildcard character (*) in the domain pattern. Use this wildcard immediately following the scheme and hierarchical portion of the domain pattern, such as http://*.domain.com to include all subdomains. The wildcard must immediately follow the scheme and hierarchical portion of the domain pattern. If you use an IP address instead of domain pattern, you must enter the full IP address without a wildcard.

Note: You cannot use multiple wildcards, or specify a wildcard without a domain pattern. Values such as * or *.* are not supported.

Domain matching

When evaluating the Origin header in a request, ServiceNow prioritizes CORS rules that match the domain pattern exactly. If no exact match is found, the next closest match is used.

For example, if there are rules for the domain patterns http://*.blog.mysite.com and http://*.mysite.com, a request from http://alice.blog.mysite.com will match the http://*.blog.mysite.com pattern.

Examples of valid and invalid domains

<table>
<thead>
<tr>
<th>Valid domain</th>
<th>Invalid domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>http://*.ms.net</td>
<td>https://*com</td>
</tr>
<tr>
<td>https://*.ms.com</td>
<td>http://*..com</td>
</tr>
<tr>
<td><a href="http://192.168.1.1">http://192.168.1.1</a></td>
<td>http://*.168.1.126</td>
</tr>
<tr>
<td>http://*.service-now.com</td>
<td><a href="http://blog.*.service-now.com">http://blog.*.service-now.com</a></td>
</tr>
<tr>
<td>http://*.com</td>
<td>http://*com</td>
</tr>
</tbody>
</table>

REST API supported headers

The REST API provides support for various headers. Some headers are mandatory for specific APIs and HTTP methods.
Data format headers

The Accept and Content-Type request headers are required for proper data formatting for requests that contain a request body or response body. These request headers support the following values:

- **Accept**: application/json, application/xml
- **Content-Type**: application/json, application/xml

**Note:** The Attachment API accepts all Content-Type values (/*). Specify the file content type when uploading an attachment. The content type is stored with file metadata allowing other tools to correctly identify and parse the file.

POST, PUT, PATCH, and DELETE operations require you to provide both headers. GET operations require only the Accept header. Failing to provide the required headers results in a 400 Bad Request error.

Other headers

All requests may contain an authentication header that specifies the user credentials to authenticate with. You can override the HTTP method, such as GET or POST, by setting the X-http-method-override header.

REST API HTTP response codes

REST Messages sent to a ServiceNow instance return a specific HTTP response code.

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Message</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Success</td>
<td>Success with response body.</td>
</tr>
<tr>
<td>201</td>
<td>Created</td>
<td>Success with response body.</td>
</tr>
<tr>
<td>204</td>
<td>Success</td>
<td>Success with no response body.</td>
</tr>
<tr>
<td>400</td>
<td>Bad Request</td>
<td>The request URI does not match the APIs in the system, or the operation failed for unknown reasons. Invalid headers can also cause this error.</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>The user is not authorized to use the API.</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
<td>The requested operation is not permitted for the user. This error can also be caused by ACL failures, or business rule or data policy constraints.</td>
</tr>
<tr>
<td>404</td>
<td>Not found</td>
<td>The requested resource was not found. This can be caused by an ACL constraint or if the resource does not exist.</td>
</tr>
<tr>
<td>405</td>
<td>Method not allowed</td>
<td>The HTTP action is not allowed for the requested REST API, or it is not supported by any API.</td>
</tr>
</tbody>
</table>
Dot-walking in REST API requests

You can use dot-walking when specifying the sysparm_query or sysparm_fields parameters in requests to REST APIs that support those parameters.

**Note**: The Import Set API does not support dot-walking.

### Dot-walking in sysparm_query

You can filter queries using related record values by dot-walking in the sysparm_query parameter. For example, you can retrieve all incident records where the incident **Company** has a specific **Stock symbol** value.

https://<instance>.service-now.com/api/now/table/incident?sysparm_query=company.stock_symbol=NYX

### Dot-walking in sysparm_fields

You can view field values from multiple tables by dot-walking in the sysparm_fields parameter. For example, you can retrieve the **Name**, **Sys_id**, and **Department** of each user that has certain roles, as well as the role **Name**.

The request runs on the User Roles [sys_user_has_role] table which defines a many-to-many relationship between users and roles. The response includes field values from the User [sys_user] and Roles [sys_user_role] tables.

https://<instance>.service-now.com/api/now/table/sys_user_has_role?sysparm_fields=role%2Crole.name%2Cuser%2Cuser.name%2Cuser.sys_id%2Cuser.department&sysparm_query=role%3D3d43716d0f6002003a2d47bce1050e0d%5EORrole%3Dac73b52d0f6002003a2d47bce1050e0d&sysparm_display_value=true

```json
{
    "result": [
        {
            "user.name": "Fred Johnson",
            "user.sys_id": "f5a3716d0f6002003a2d47bce1050ed4",
            "role.name": "support",
            "user.department": {
                "display_value": "Accounting",
                "link": "https://<instance>.service-now.com/api/now/table/cmn_department/5b3b1530f58c2003a2d47bce1050e96"
            },
            "role": {
                "display_value": "support",
                "link": "https://<instance>.service-now.com/api/now/table/sys_user_role/3d43716d0f6002003a2d47bce1050e0d"
            }
        }
    ]
}
```
Debugging REST queries

You can debug REST queries by reviewing the session debug log.

When the property glide.rest.debug is true, all REST processing is logged in the session debug log.

REST Logging includes processing durations, headers, and the request body. Prolonged use of this property can affect performance, so it is best to use it while debugging REST processing, and then set the property back to false.

Sample log output

2014-03-19 11:10:37 (633) http-12 New transaction
083A6031D7231100261253B2B252035C #28 /api now//table/incident
2014-03-19 11:10:37 (653) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Started initializing REST Request
2014-03-19 11:10:37 (653) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Request Method:POST
2014-03-19 11:10:37 (653) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Request Header: host:localhost:8080
2014-03-19 11:10:37 (656) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Request Header: user-agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:12.0) Gecko/20100101 Firefox/12.0
2014-03-19 11:10:37 (656) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Request Header: accept:application/json
2014-03-19 11:10:37 (656) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Request Header: accept-encoding: gzip, deflate
2014-03-19 11:10:37 (656) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Request Header: connection:keep-alive
2014-03-19 11:10:37 (657) REST API-thread-1 SYSTEM DEBUG: [REST API]
RESTAPIProcessor : Request Header: content-type:application/json; charset=UTF-8
2014-03-19 11:10:37 (657) REST API-thread-1 SYSTEM DEBUG: [REST API] RESTAPIProcessor : Request Header: cookie:glide_user_route=glide.20e7f4c6d6d0d44810117aacc0eae;
JSESSIONID=F07CE6AFC8AF237CB39AF43BF360BF;
glide_user="U0N2Mjo0MDNhNyAzMWQ3MjMxMTAwMyYjYiMjUyMDM2OT02ODE2Zjc5Y2MwYTgweiM0MDM3
2014-03-19 11:10:38 (357) REST API-thread-1 SYSTEM DEBUG: [REST API] RouteRegistry : Loaded Routes to Cache
2014-03-19 11:10:38 (357) REST API-thread-1 SYSTEM DEBUG: [REST API] RouteRegistry : Route loading time 0:00:00.105
-------------------------
api=api
2014-03-19 11:10:38 (541) REST API-thread-1 083A6031D7231100261253B2B252035C DEBUG: [REST API] RESTAPIProcessor : Pre-Service processing duration 0:00:00.000
2014-03-19 11:10:39 (508) REST API-thread-1 083A6031D7231100261253B2B252035C DEBUG: [REST API] TableAPIService : Glide Record Insert Duration 0:00:00.956

© 2017 ServiceNow. All rights reserved.
Scripted REST APIs

The Scripted REST APIs feature allows application developers to build web service APIs.

You can define service endpoints, query parameters, and headers for a scripted web service API, as well as scripts to manage the request and response.

Scripted REST APIs generally follow the REST architecture, but may be customized to use different conventions.

The following podcast offers additional information on the use of Scripted REST APIs.

Scripted REST core concepts

Review the Scripted REST APIs core concepts before creating your own APIs.

**Scripted REST API URIs**

Scripted REST API URIs depend on the namespace and API ID, and optionally the version, of the web service.

The API URI, excluding the instance URL, appears in the Resource path and Default resource path fields for scripted REST resource records. The Default resource path field appears only for resources with the default API version.

URIs used by Scripted REST APIs follow these formats:

- `https://instance.service-now.com/api/(name_space)/(api_id)/(resource_path)`
- `https://instance.service-now.com/api/(name_space)/(version)/(api_id)/(resource_path)`

In these URIs:

- `(name_space)` is a read-only value. For web services in the global scope, the name space is the value of the property glide.appcreator.company.code. For web services in a scoped application, the name space is the scope name, such as `x_company_appname`.

© 2017 ServiceNow. All rights reserved.
• (api_id) is the value of the API ID field on the scripted REST API record. By default this value is based on the service name.

• (resource_path) is the relative path defined for the resource. Specifying a relative resource path allows you to have multiple resources using the same HTTP method, such as GET, in one web service. For example, an resource may specify the path /{id} when the web service has only one GET resource, or /user/{id} and /message/{id} when the web service has different resources for requesting user and message records.

• (version) is the version of the API to access if the API uses versioning, such as v1. You can access the default version of a versioned API by using the URI without a version number.

**Scripted REST API roles**

Certain roles are required to work with scripted REST API records. Roles required to access each API depend on the API implementation.

The following roles are required to work with scripted REST API records.

These roles are not required to access a scripted REST API endpoint. The REST API author can define security requirements for each API and API resource.

<table>
<thead>
<tr>
<th>Role title [name]</th>
<th>Description</th>
<th>Contains Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web service administrator [web_service_admin]</td>
<td>Users with this role can read, create, modify, and delete Scripted REST APIs and web service resources.</td>
<td>none</td>
</tr>
</tbody>
</table>

**Scripted REST API query parameters**

Query parameters define values that requesting users can pass with a request.

When creating a scripted REST API you can specify which parameters are available and which are mandatory for each request. You can access request parameters in scripted REST API scripts using the request object queryParams field.

**Scripted REST API versioning**

Scripted REST APIs may be versioned, allowing you to test and deploy changes without impacting existing integrations.

**Enable versioning**

By default, new scripted REST APIs are not versioned. To enable versioning, click Enable versioning on the Scripted REST API form.

**Note:** To continue supporting non-versioned URLs after enabling versioning, select a version as the default version.

**Default version**

A version may be marked as default. Specifying a default version allows users to access that version using a web service URL without a version number.

If more than one active version is marked as default, the latest default version is used as the default.
**Add a version**

To add a new version to a scripted REST service, click **Add new version** on the Scripted REST API form.

When you add a new version, you can copy resources from an existing version.

**Scripted REST API access controls**

Scripted REST APIs use REST_Endpoint type ACLs to define access controls.

As a user with the web_service_admin role, you can define access controls for scripted REST APIs and scripted REST API resources.

A scripted REST API or scripted REST resource may require users to submit basic authentication credentials and pass an ACL check. A requesting user must satisfy at least one of the ACLs. It is not necessary to satisfy all selected ACLs.

Scripted REST service ACLs must have the **Type** value **REST_Endpoint**.

**Scripted REST API security matrix**

There are multiple possible security configurations for scripted REST APIs.

Use this table to identify the scripted REST API security configuration that best suits your needs, and the field values to implement that configuration.

### Table 1022: Scripted REST API security

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Scripted REST API</th>
<th>Scripted REST Resource</th>
<th>Requires authentication</th>
<th>Requires ACL authorization</th>
<th>ACLs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default ACLs</strong></td>
<td>Any value</td>
<td>False</td>
<td>Any value</td>
<td>Any value</td>
<td></td>
</tr>
<tr>
<td>The resource is public. No authentication or ACL is required.</td>
<td>Any value</td>
<td>True</td>
<td>False</td>
<td>Any value</td>
<td></td>
</tr>
<tr>
<td>The resource requires basic authentication only. No ACL is required.</td>
<td>No ACL selected</td>
<td>True</td>
<td>True</td>
<td>No ACL selected</td>
<td></td>
</tr>
<tr>
<td>The resource requires basic authentication only. No ACL is required.</td>
<td>Any value</td>
<td>True</td>
<td>True</td>
<td>One or more ACLs selected</td>
<td></td>
</tr>
<tr>
<td>An ACL selected in the resource record is required.</td>
<td>One or more ACLs selected</td>
<td>True</td>
<td>True</td>
<td>No ACL selected</td>
<td></td>
</tr>
</tbody>
</table>

**Scripted REST API error objects**

Scripted REST APIs include error objects that allow you to respond to a request with a standard HTTP error message when an error occurs during request processing.
You can use error objects in scripted REST API resources to alert requesting clients of errors. Use error objects to respond to incoming requests, not to catch errors within your server-side code.

## Error response format

The content type of the response depends on the request Accept header. If the Accept header specifies an unsupported format, such as image/jpeg, the error response uses JSON.

Error responses follow this format:

```
{
   "error": {
      "message": "My error message",
      "detail": "My details"
   },
   "status": "failure"
}
```

The numeric status code, such as 404, is included in the response Status code header, not in the response body.

### Sending scripted REST API errors

Scripted REST APIs provide multiple ways to send an error in a response to a requesting client. The **Script** field on the Scripted REST Resource form provides a process function template. You can send an error response by returning an error object, or using the process function response parameter. Either option returns the same error to the requesting client.

### Returning an error

You can send an error by returning an error object.

```
(function process(request, response) {
   // request is bad
   return new sn_ws_err.BadRequestError('My error message');
})(request, response);
```

### Setting a response error

You can send an error by calling the `setError` function on the response parameter.

```
(function run(request, response) {
   // request is bad
   responseBuilder.setError(new sn_ws_err.BadRequestError()); // Constructor parameter is optional
})(request, response);
```

### Available error objects

Multiple error objects are available in scripted REST API scripts to report error information to requesting clients.
Error namespace

All scripted REST API error objects use the sn_ws_err namespace. When instantiating a new error object, use this namespace, such as in `var error = new sn_ws_err.BadRequestError()`.

Predefined errors

Predefined error objects allow you to easily communicate an error based on standard HTTP error codes. You can set an optional error message in the object constructor.

<table>
<thead>
<tr>
<th>Error Object</th>
<th>Description</th>
<th>Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>BadRequestError</td>
<td>Indicates an error in the request, such as incorrect syntax.</td>
<td>400</td>
</tr>
<tr>
<td>NotFoundError</td>
<td>Indicates a requested resource is not available.</td>
<td>404</td>
</tr>
<tr>
<td>NotAcceptableError</td>
<td>Indicates the Accept header value passed in the request is incompatible with the web service.</td>
<td>406</td>
</tr>
<tr>
<td>ConflictError</td>
<td>Indicates that there is a conflict in the request, such as multiple conflicting updates.</td>
<td>409</td>
</tr>
<tr>
<td>UnsuppotedMediaTypeError</td>
<td>Indicates the request media type is not supported by the web service.</td>
<td>415</td>
</tr>
</tbody>
</table>

```javascript
(function run(request, response) {
  // request is bad
  return new sn_ws_err.BadRequestError('My error message');
})(request, response);
```

Customizable ServiceError

The ServiceError object allows you to define your own error response. You can set the status code, error message, and error details using a ServiceError object. Use the ServiceError object when none of the predefined errors meet your needs.

The ServiceError object exposes these functions for setting values:
Table 1024: Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>setStatus(Number status)</td>
<td>Use this function to set the numeric status code for the error.</td>
<td>500</td>
</tr>
<tr>
<td>setMessage(String message)</td>
<td>Use this function to set a short error message.</td>
<td>An empty string</td>
</tr>
<tr>
<td>setDetail(String detail)</td>
<td>Use this function to set the detailed error description.</td>
<td>An empty string</td>
</tr>
</tbody>
</table>

```javascript
(function run(request, response) {
    var myError = new sn_ws_err.ServiceError();
    myError.setStatus(410);
    myError.setMessage('My error Message');
    myError.setDetail('Here are the details about this error');
    return myError;
})(request, response);
```

Create a scripted REST API

Create a scripted REST API to define web service endpoints.

Role required: web_service_admin

1. Navigate to **System Web Services > Scripted REST APIs**.
2. Click **New**.
3. Enter a **Name** for the service.
   The **API ID** is set automatically based on the **Name**. You can modify the **API ID** if needed.
4. Click **Submit**.

After you create the API, configure the service as needed such as by creating resources, assigning ACLs, or specifying supported request and response formats.

*Create a scripted REST API resource*

Create a scripted REST API resource to define the HTTP method, the processing script, and to override settings from the parent service.

There must be a Scripted REST API defined before you can create resources.

Role required: web_service_admin

1. Navigate to **System Web Services > Scripted REST APIs**.
2. Select a scripted REST API record.
3. In the **Resources** related list, click **New**.
4. Enter a **Name**.
   The resource name affects the URI for sending requests to the API.
5. Select the **HTTP method** this resource implements, such as **GET**.
6. In the **Script** field, define how the operation parses and responds to requests.
7. Optional: Override settings from the parent REST API as needed, such as by specifying different security settings or valid content types.
8. Optional: On the **Documentation** tab, provide a **Short description** explaining how to access the resource.
   
   This information appears when exploring this resource using the REST API Explorer.

9. Click **Submit**.

After creating the resource, you can associate headers and query parameters.

**Define scripted REST API headers**

Define scripted REST API headers to control which headers the API accepts and can respond with.

There must be a Scripted REST API defined before you can create headers.

Role required: web_service_admin

1. Navigate to **System Web Services > Scripted REST APIs**.
2. Select a scripted REST API record.
3. In the **Request Headers** related list, click **New**.
4. Enter a **Header name**.
5. Optional: Select **Is required** to make this header mandatory for all requests to associated scripted REST resources.
6. Enter a **Short description** and **Example value** to explain how to use the header.
7. Click **Submit**.

After defining available headers, associate the headers with a scripted REST resource.

**Define available query parameters**

Define available query parameters to control what values a requesting user can pass in the request URI.

Role required: web_service_admin

1. Navigate to **System Web Services > Scripted REST APIs**.
2. Select a scripted REST API record.
3. In the **Query Parameters** related list, click **New**.
4. Specify the **Query parameter name**.
5. Optional: Select **Is required** to make this parameter mandatory for all requests to associated scripted REST resources.
6. Enter a **Short description** and **Example value** to explain how to use the parameter.
7. Click **Submit**.

After defining available query parameters, associate the parameters with a scripted REST resource.

**Configure a scripted REST API to require an ACL**

Configure a scripted REST API to require that requesting users satisfy an ACL.

Role required: web_service_admin

Requests to scripted REST APIs respect platform ACLs. The requesting user must meet any table ACL requirements to access instance data. Additionally, you can configure the scripted REST API to require a specific ACL.

1. Navigate to **System Web Services > Scripted REST APIs**.
2. Select a scripted REST API record.
3. In the **Default ACL** field, select one or more ACLs that have a **Type** value of **REST_Endpoint**.
   
   A requesting user must satisfy at least one of the selected ACLs. It is not necessary to satisfy all selected ACLs.
4. Click **Update**.

You can override the API security settings for each individual resource.
Configure a scripted REST API resource to require an ACL

As a user with the web_service_admin role, you can configure a scripted REST API resource to require ACLs.

Role required: web_service_admin or admin

Resources inherit security settings from the parent API by default. Define custom security settings for a specific resource to override the inherited settings.

1. Navigate to System Web Services > Scripted REST APIs.
2. Select a scripted REST API record.
3. In the Resources related list, select a resource.
4. In the Security tab, select the Requires authentication check box.
   
   This check box must be selected to require an ACL. Clear this check box to allow unauthenticated requests to access the resource, even if the parent REST service requires an ACL.
5. Select the Requires ACL authorization check box.
6. In the ACL field, select one or more ACLs that have a Type value of REST_Endpoint.
   
   Selecting an ACL for an resource overrides any ACLs selected for the parent web service. Leave this field blank to use the ACLs selected for the parent web service.

Enable versioning for a scripted REST API

Enable versioning for a scripted REST API to provide multiple versions of the API while maintaining compatibility with existing integrations.

There must be a Scripted REST API defined before you can enable versioning.

Role required: web_service_admin

1. Navigate to System Web Services > Scripted REST APIs.
2. Select a scripted REST API record.
3. Click Enable versioning.
   
   The Enable versioning pop up appears. The Make version v1 default check box is selected by default.
4. Optional: Clear the Make version v1 default check box to enable versioning without a default version.
   
   Versioned APIs without a default version are accessible only by using the version-specific URI. Make version v1 default, or select a different version as default after you enable versioning.
5. Click OK.
   
   The Versioning embedded list is added to the Scripted REST API form. You can add new versions or control which version is default from this list.

Add a version to a scripted REST API

Add a new version to a versioned scripted REST API to define new API behavior without impacting older versions.

There must be a Scripted REST Service that has versioning enabled before you can add a new version.

Role required: web_service_admin

1. Navigate to System Web Services > Scripted REST APIs.
2. Select a scripted REST service record.
3. Click Add new version.
   
   The Add new version pop up appears.
4. Optional: Select Make this version the default to configure the REST service to use the new version as the default version.
5. Optional: In the Copy existing resources from version choice list, select an existing version to copy all resources from that version to the new version.
6. Click OK.

Controlling request and response content type
You can control which content types are allowed in scripted REST API requests and responses.

You can set default values for the service using the **Default supported request formats** and **Default supported response formats** fields. These fields define acceptable values users can pass in the Content-Type and Accept request headers respectively. If a requesting user specifies an Accept or Content-Type header not supported by the service or resource, the instance responds with an HTTP error code 406 or 415.

You can override these values for each resource using the **Supported request formats** and **Supported response formats** on the Scripted REST Resource form.

**Note:** The **Supported request formats** field only appears for PUT, POST, and PATCH resources.

You can use wildcard values when specifying valid content types.

If no accepted content types are specified, scripted REST web services support Application/json, Application/xml, and Text/xml by default.

**Important:** If the request body format is not Application/json, Application/xml, or Text/xml, use only the request body `dataStream` field to access the request body. Using request body `data`, `dataString`, `nextEntry()`, or `hasNext()` with a non-default format will result in a 500 error response.

When sending a binary type in a response, you must set the response content type and write the binary stream directly using a RESTAPIResponseStream object. You can access this object by calling `getStreamWriter()` on the response object.

Controlling maximum request size
You can specify the maximum file size allowed in a scripted REST API request payload.

The file size limit applies when accessing any of the following variables or functions from a RESTAPIRequestBody object.

- `data`
- `dataString`
- `nextEntry()`
- `hasNext()`

Accessing these variables or functions with a request payload that exceeds the maximum size causes the service to respond with error code 400.

The file size limit does not apply when accessing the `dataStream` variable.

**Maximum request size properties**

Several properties control the maximum allowed request size. Add these properties to the System Properties [sys_properties] table to specify a maximum request size.
### Explore scripted REST APIs

You can browse scripted REST APIs using the REST API Explorer and view documentation provided by scripted REST API creators.

The REST API must be active and must specify at least one resource to appear in the REST API Explorer.

You can provide additional information about scripted REST APIs, API resources, request headers, and query parameters. The **Short description** field from these records appears on the REST API Explorer when viewing the appropriate API and resource. You can also provide a **Documentation link** URL in the scripted REST API record to direct users to more detailed documentation. Users can access this URL by clicking **API documentation** in the REST API Explorer.

### Scripted REST APIs good practices

Follow these guidelines when designing and implementing Scripted REST APIs.

**Follow REST API conventions**

Use REST API standards to provide a consistent and easy to use interface for clients.

REST API conventions define specific behavior for each type of method. Use the following guidelines as a starting point for designing your API.

- **GET** operations only query data. A GET request should never modify data.
- **POST** operations create new records but do not modify existing records.
- **PUT** and **PATCH** operations modify existing records.
- **DELETE** operations destroy records.

**Use versioning to control changes to your API**

Use versioning to implement new functionality and avoid breaking existing integrations.

When you introduce significant functionality changes to your API, create a new version of the API first. Do not introduce behavior that will break existing integrations in a published version.

Using versioning allows you to implement significant changes to your API without breaking existing clients. You can then release the new version of the API for new clients while allowing existing clients to upgrade at their own pace.

Encourage clients to use a version-specific API, or configure the API without a default version to force clients to specify a version. You can also make new, optional behavior available by adding an optional parameter to an existing version.

**Return an informative HTTP status code**

Return a status code that informs the requestor of the success or failure of the request.

Return an HTTP status code that helps the client understand the result of the request. Use the following guidelines for common status codes.

---

**Table 1025: Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.rest.scripted.max_inbound_content_length_mb</td>
<td>10</td>
<td>The maximum size, in megabytes, for a scripted REST request body that is not gzipped.</td>
</tr>
<tr>
<td>glide.rest.scripted.max_inbound_gzip_content_length_mb</td>
<td>1</td>
<td>The maximum size, in megabytes, for a scripted REST request body that is gzipped.</td>
</tr>
</tbody>
</table>
Table 1026: Common status codes

<table>
<thead>
<tr>
<th>Status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Indicates that the request was completed successfully.</td>
</tr>
<tr>
<td>201</td>
<td>Indicates that a record was created successfully.</td>
</tr>
<tr>
<td>204</td>
<td>Indicates that a record was deleted successfully.</td>
</tr>
<tr>
<td>40X (401, 404)</td>
<td>Status codes in the 400 range indicate a client error, such as 400 for invalid request syntax.</td>
</tr>
<tr>
<td>50X (500, 503)</td>
<td>Status codes in the 500 range indicate that a server error occurred. The client request may have been valid or invalid, but a problem occurred on the server that prevented it from processing the request.</td>
</tr>
</tbody>
</table>

*Return useful error information*

Provide the client with enough information in error messages to allow them to understand the problem without having to refer to your API documentation.

An error response should include a helpful error message, as well as an error status code.

For example, when a client queries a record that does not exist, you can return the error message “The specified record does not exist. Ensure that a record with the ID of <id value> exists in the application.” along with a 404 status code.

The Scripted REST APIs feature includes several preconfigured error objects you can use for commonly-encountered errors, and a customizable ServiceRequest error object you can use when the preconfigured error objects do not meet your needs.

*Enforce and test access controls*

Enforce existing access controls and require additional access to modify data.

In addition to requiring authentication to access the API, require authorization to access data. Use the GlideRecordSecure API in your Scripted REST API scripts. This API ensures that access controls defined on the underlying data are applied for the requesting user.

Require additional access controls for operations that modify data. Requests such as PUT, POST, and DELETE should require a higher level of access than GET. Configure these API resources to require a more strict ACL.

Test your access controls, both authentication and authorization, before releasing the API.

*Build tests to verify functionality*

Build tests that verify your Scripted REST web services functionality as part of your development process.

Use repeatable tests to ensure that your API functions the way you expect it to. Testing also helps ensure that changes you make do not affect the expected API behavior after you release a version. You can use a REST client application that supports automated testing, such as Postman, to facilitate testing.

Tests should validate the response code, headers, and body content as appropriate for each resource you implement. You can also use tests to validate authentication requirements, and to confirm that errors return useful responses.

RESTAPIRequest

A RESTAPIRequest object allows you to access scripted REST API request details in scripts.
**Note:** You cannot instantiate objects of this type. Objects of this type are created automatically and are accessible only in scripted REST API resource scripts.

**RESTAPIRequest - body**

The body of the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>RESTAPIRequestBody</td>
<td>The body of the request. You can access data from the body object using the RESTAPIRequestBody API.</td>
</tr>
</tbody>
</table>

```javascript
var requestBody = request.body; // Returns instance of RESTAPIRequestBody
```

**RESTAPIRequest - pathParams**

The path parameters passed in the request URI.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pathParams</td>
<td>Object</td>
<td>The path parameters as a script object. Available path parameters depend on the web service configuration.</td>
</tr>
</tbody>
</table>

In this example, the scripted REST API endpoint follows this format: `https://instance.service-now.com/api/now/myservice/{tableName}/{id}`. The request being processed uses this URL: `https://instance.service-now.com/api/now/myservice/myApp_table/1234`.

```javascript
var pathParams = request.pathParams;
var tableName = pathParams.tableName; // ‘myApp_table’
var id = pathParams.id; // ‘1234’
```

**RESTAPIRequest - queryParams**

The query parameters from the web service request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryParams</td>
<td>Object</td>
<td>The query parameters from the web service request.</td>
</tr>
</tbody>
</table>

```javascript
```
In this example, the request being processed uses this URL: https://<instance_rest_endpoint>?active=false&name=now. Note the active and name parameters.

```javascript
var queryParams = request.queryParams;
var isActiveQuery = queryParams.active; //false
var nameQueryVal = queryParams.name; //"now"
```

RESTAPIRequest - queryStr
The entire query added to the endpoint URI.

### Table 1030: Field

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryString</td>
<td>String</td>
<td>The entire query for the request.</td>
</tr>
</tbody>
</table>

In this example, the request being processed uses this URL: https://<instance_rest_endpoint>?active=false&name=now. Note the query active=false&name=now.

```javascript
var query = request.queryString; //"active=false&name=now"
```

RESTAPIRequest - uri
The request URI, excluding domain information.

### Table 1031: Field

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uri</td>
<td>String</td>
<td>The request URI, excluding domain information.</td>
</tr>
</tbody>
</table>

In this example, the request being processed uses this URL: https://instance.service-now.com/api/now/table/myTable?active=false&name=now.

```javascript
var query = request.uri; //"api/now/table/myTable"
```

RESTAPIRequest - url
The entire request URL.

### Table 1032: Field

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>String</td>
<td>The entire request URL.</td>
</tr>
</tbody>
</table>
In this example, the request being processed uses this URL:
> https://instance.service-now.com/api/now/table/myTable?
> active=false&name=now.

```javascript
var query = request.url; //"https://instance.service-now.com/
> api/now/table/myTable?active=false&name=now"
```

**RESTAPIRequest - headers**
All headers from the request.

### Table 1033: Field

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>headers</td>
<td>object</td>
<td>All headers from the request, and their values.</td>
</tr>
</tbody>
</table>

```javascript
var headers = request.headers;
var acceptHeader = headers.Accept;
var myCustomHeader = headers.myCustom;
var specialHeader = headers['special - header'];
```

**RESTAPIRequest - getHeader(String header)**
Get the value of a specific header from the web service request.

### Table 1034: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>header</td>
<td>String</td>
<td>The name of the header, such as accept or content-type.</td>
</tr>
</tbody>
</table>

### Table 1035: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The value of the specified header.</td>
</tr>
</tbody>
</table>

```javascript
var acceptHeader = request.getHeader('accept');
```

**RESTAPIRequest - getSupportedResponseContentTypes()**
Get the content types specified in the request Accept header.
### Table 1036: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 1037: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>An array of string values where each string is a content type, such as application/json.</td>
</tr>
</tbody>
</table>

**Output:**

**RESTAPIRequestBody**

A RESTAPIRequestBody object allows you to access the body content of a scripted REST API request in scripts.

The format of a RESTAPIRequestBody object may be JSON or XML, depending on the content-type header value from the request.

**Note:** You cannot instantiate objects of this type. Objects of this type are created automatically and are accessible only in scripted REST API resource scripts.

Single entry example-request-body in JSON format.

```json
{
   "name": "user1",
   "id": 1234,
   "roles": [
      {
         "name": "admin"
      },
      {
         "name": "itil"
      }
   ]
}
```

Multiple entry example-request-body in JSON format.

```json
[{
   "name": "user1",
   "id": 1234,
   "roles": [
      {
         "name": "admin"
      },
      {
         "name": "itil"
      }
   ]
}]
```
Important: If the request body format is not Application/json, Application/xml, or Text/xml, use only the request body `dataStream` field to access the request body. Using request body `data`, `dataString`, `nextEntry()`, or `hasNext()` with a non-default format will result in a 500 error response.

**RESTAPIRequestBody - data**
The content of the request body.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Object or Array</td>
<td>The request content. This can be a single object or an array of objects depending on the request.</td>
</tr>
</tbody>
</table>

```javascript
var entry;
var id;
var requestBody = request.body;
var requestData = requestBody.data; // May be an array or a single object
if (requestData instanceof Array) {
  entry = requestData[0].name; // 'user1'
  id = requestData[0].id; // '1234'
} else {
  entry = requestData.name; // 'user1'
  id = requestData.id; // '1234'
}
```

**RESTAPIRequestBody - dataStream**
The content of the request body, as a stream.
Table 1039: Field

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataStream</td>
<td>Object</td>
<td>The content of the request body. You can pass the stream to a separate API, such as to create an attachment from the request or forward the request to a different endpoint.</td>
</tr>
</tbody>
</table>

```javascript
var requestBody = request.body;
var requestString = requestBody.dataStream;
```

**RESTAPIRequestBody - dataString**
The content of the request body, as a String.

Table 1040: Field

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataString</td>
<td>String</td>
<td>The content of the request body.</td>
</tr>
</tbody>
</table>

```javascript
var requestBody = request.body;
var requestString = requestBody.dataString;
```

**RESTAPIRequestBody - hasNext()**
Determine if there are additional entries in the request body.
Use this method with the nextEntry() method to iterate over multiple request body entries.

Table 1041: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1042: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>True if there are additional entries available. This method returns true only once if the request contains a single entry.</td>
</tr>
</tbody>
</table>

```javascript
var requestBody = request.body;
```
requestBody.hasNext(); // returns true if the request contains a single entry or multiple entries

// calling second time
requestBody.hasNext(); // returns false if the request contains a single entry, or true if the request contains multiple entries

**RESTAPIRequestBody - nextEntry()**
Retrieve one entry from the request body as a script object.
Use this method with the hasNext() method to iterate over multiple request body entries.

**Table 1043: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1044: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>A single entry from the request body.</td>
</tr>
</tbody>
</table>

```
var requestBody = request.body;
var requestEntry = requestBody.nextEntry(); // returns available entry if there is only one entry, or the first entry if there are multiple.
var name = requestEntry.name; // ‘user1’

// Calling second time
requestEntry = requestBody.nextEntry(); // returns undefined if there is only one entry, or the second entry if there are multiple.
```

This example demonstrates using hasNext() with nextEntry().

```
var requestBody = request.body;
while(requestBody.hasNext()){
    var entry = requestBody.nextEntry();
}
```

**RESTAPIResponse**

A RESTAPIResponse object allows you to build a RESTful response to a scripted REST API request.

**Note:** You cannot instantiate objects of this type. Objects of this type are created automatically and are accessible only in scripted REST API resource scripts.
RESTAPIResponse - `getStreamWriter()`
Get the ResponseStreamWriter for this response, allowing you to write directly to the response stream.

Set the content type and status code using the `setHeaders` and `setStatus` functions prior to calling the `getStreamWriter` function. If you do not set a status code, the status code defaults to 200.

### Table 1045: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 1046: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTAPIResponseStream on page 3243</td>
<td>The ResponseStreamWriter for this response. You can use this object to write directly to the response stream.</td>
</tr>
</tbody>
</table>

```javascript
response.setContentType('application/json');
response.setStatus(200);
var writer = response.getStreamWriter();
```

RESTAPIResponse - `setBody(Object body)`
Set the body content to send in the web service response.

### Table 1047: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>Object</td>
<td>The response body, as a JavaScript object. The body content is automatically serialized to JSON or XML depending on the value of the Accept header passed in the request.</td>
</tr>
</tbody>
</table>

### Table 1048: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var body = {};
body.name = "incident";
body.number = "1234";
body.caller = {"id": "user1"};
```
response.setBody(body);
var bodyArray = [];
var body = {};
body.name = "incident";
body.number = "1234";
body.caller = {"id":"user1"};
bodyArray.push(body);
response.setBody(bodyArray);

**RESTAPIResponse - setHeaders(Object headers)**
Set the headers for the web service response.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>headers</td>
<td>Object</td>
<td>A JavaScript object listing each header and the value to assign that header.</td>
</tr>
</tbody>
</table>

**RESTAPIResponse - setLocation(String location)**
Assign a value to the Location header in the web service response.

See the *W3 Location header documentation* for more information about this header.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>String</td>
<td>An absolute URI to redirect the response recipient to.</td>
</tr>
</tbody>
</table>

Table 1050: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>
RESTAPIResponse - setStatus(Number status)
Set the status code number for the web service response.

Table 1053: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Number</td>
<td>The status code to send in the response, such as 200 to indicate success. Passing a non-numerical value, such as a string, causes the status code to default to 0.</td>
</tr>
</tbody>
</table>

Table 1054: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```
response.setStatus(200);
```

RESTAPIResponse - setHeader(String header, String value)
Assign a value to a REST service response header.

Table 1055: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>header</td>
<td>String</td>
<td>The header you want to set.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign the specified header.</td>
</tr>
</tbody>
</table>

Table 1056: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```
responseBuilder.setHeader("Location","<URI>");
```
RESTAPIResponse - contentType(String contentType)
Assign a value to the Content-Type header in the web service response.

You must set a response content type before writing the response. The content type is set automatically for string responses, based on the request Accept header value.

Setting an invalid content type causes the response to default to JSON. Failing to set a content type results in a status code 500 error when sending a binary response.

See the W3 Content-Type header documentation for more information about this header.

**Table 1057: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contentType</td>
<td>String</td>
<td>The content type of the response body, such as application/json.</td>
</tr>
</tbody>
</table>

**Table 1058: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
responseBuilder.setContentType('application/json');
```

RESTAPIResponse - setError(Object error)
Configure the response to return an error.

**Table 1059: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>Object</td>
<td>An error object.</td>
</tr>
</tbody>
</table>

**Table 1060: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

Output:

RESTAPIResponseStream
A RESTAPIResponseStream object allows you to write directly to the scripted REST API response stream.
**Note:** You cannot instantiate objects of this type. Objects of this type are created automatically and are accessible only in scripted REST API resource scripts.

RESTAPIResponseStream - writeStream(Object stream)
Write an input stream to the response stream.

You must set the content type and status code before calling the `writeStream()` method or the response will fail. You cannot modify these values after calling the `writeStream()` method.

### Table 1061: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stream</td>
<td>Object</td>
<td>An attachment or a response stream from a third-party service.</td>
</tr>
</tbody>
</table>

### Table 1062: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

It is the responsibility of the script author to obtain the stream from a third-party service.

```java
response.setContentType('application/json');
response.setStatus(200);
var writer = response.getStreamWriter();
var attachmentStream = new GlideSysAttachmentInputStream(<sys id of attachment>);
writer.writeStream(attachmentStream);
```

RESTAPIResponseStream - writeString(String data)
Write string data to the response stream.

You must set the content type and status code before calling the `writeString()` method or the response will fail. You cannot modify these values after calling the `writeString()` method.

### Table 1063: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>String</td>
<td>The string to add to the response data.</td>
</tr>
</tbody>
</table>
Table 1064: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
response.setContentType('application/json');
response.setStatus(200);
var writer = response.getWriter();
var body = {
  name: 'user1',
  id: 1234,
  roles: [
    { name: 'admin' },
    { name: 'itil' }
  ]
};
writer.writeString('{"name":"user","id":"1234"}');
writer.writeString(JSON.stringify(body));
```

Scripted REST API examples

Multiple examples are available demonstrating how to create and use scripted REST APIs.

*Scripted REST API example - script samples*

These examples demonstrate how to create various resource scripts for a scripted REST API.

**Query parameters GET example**

This example demonstrates how to get query parameter values from a request.

```javascript
/**
 * GET - Sample Request API - Query Params
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
  var uri = request.uri;
  var url = request.url;
  var queryParams = request.queryParams;
  var customHeader = request.getHeader('X-Custom');

  return {
    "uri": uri,
    "url": url,
    "queryParams": queryParams,
    "customHeader": customHeader
  };
})(request, response);
```
Path parameters GET example

This example demonstrates how to get path parameter values from a request.

```javascript
/**
 * GET - Sample Request API - Path Params
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
    var uri = request.uri;
    var url = request.url;
    var path = request.pathParams;

    return {
        "uri": uri,
        "url": url,
        "path_params": path,
        "path.id": path.id
    };
})(request, response);
```

Script include GET example

This example demonstrates how to use a script include to provide a response. By using a script include you can reuse common code and maintain readability in the REST service scripts.

```javascript
/**
 * GET - Sample Request API - Script Include
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
    var responseObj = global.SampleDataUtil.getSampleJSON();
    return responseObj;
})(request, response);
```

String POST example

This example demonstrates how to parse a POST message with a string body and send a response based on the request.

```javascript
/**
 * POST - Sample Request API - dataString
 * sample usage:
 * var requestBody = request.body;
 * var requestString = requestBody.dataString;
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
    var requestBody = request.body;
    var requestString = requestBody.dataString;
    return {"requestString": requestString};
})(request, response);
```
**Binary POST example**

This example demonstrates how to parse a POST message with a binary body and send a response based on the request.

```javascript
/**
 * POST - Sample Request API - Body
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response)
{
    var body = request.body.data;
    //do any additional processing on the request body, such as inserting a new record.
    return {
       "body.id": body.id
    }
})(request, response);
```

**Not acceptable error example**

This example demonstrates how to respond with a not acceptable error. Use this error type when the request Accept header value is not supported by the web service.

```javascript
/**
 * Sample Not Acceptable Error Sample
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response)
{
    response.setError(new sn_ws_err.NotAcceptableError('sample error message'));
})(request, response);
```

**Bad request error example**

This example demonstrates how to respond with a bad request error. Use this error type to indicate a mistake in the request syntax.

```javascript
/**
 * Bad Request Error Sample
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response)
{
    response.setError(new sn_ws_err.BadRequestError('sample error message'));
})(request, response);
```

**Conflict error example**

This example demonstrates how to respond with a conflict error. Use this error type in the event of multiple conflicting requests, such as multiple updates to the same record.

```javascript
/**
 * Error Response: Conflict Error Sample
 */
```
Not found error example

This example demonstrates how to respond with a not found error. Use this error type if the requested resource does not exist or is unavailable.

```javascript
/**
 * Error Response: Not Found Error Sample
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response)
 {
   response.setError(new sn_ws_err.NotFoundError('sample error message'));
 })(request, response);
```

Unsupported media type error example

This example demonstrates how to respond with an unsupported media type error. Use this error type to indicate that the Content-Type of the request is unsupported.

```javascript
/**
 * Error Response: Unsupported Media Type Error Sample
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response)
 {
   response.setError(new sn_ws_err.UnsupportedMediaTypeError('sample error message'));
 })(request, response);
```

Service error example

This example demonstrates how to respond with a generic service error. The ServiceError object allows you to define the status code, message, and error detail. Use a ServiceError if the predefined error types do not meet your needs.

```javascript
/**
 * Error Response: Custom Error Sample
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response)
 {
   var myError = new sn_ws_err.ServiceError();
   myError.setStatus(418);
   myError.setMessage("I am a Teapot");
   myError.setDetail("Here are the details about this error");
   response.setError(myError);
 })(request, response);
```
Scripted REST resource script example

This sample REST API resource script parses the name and id values from the request body and returns those values in the response.

```javascript
/**
 * POST - Sample Request API - Body
 */
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
  var body = request.body.data,
  id0,name0,id1,name1;
  name0 = body[0].name; // 'user0'
  id0 = body[0].id; // '1234'
  name1 = body[1].name; // 'user1'
  id1 = body[1].id; // '5678'

  return {
    "id": id0,
    "name": name0,
    "id1": id1,
    "name1": name1
  }
})(request, response);
```

Requests

The API can accept both XML and JSON requests.

<table>
<thead>
<tr>
<th>JSON Request</th>
<th>XML Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST /api/sn_demo_api/v1/example/ body HTTP/1.1</td>
<td>POST /api/sn_demo_api/v1/example/ body HTTP/1.1</td>
</tr>
<tr>
<td>Content-Type: application/json</td>
<td>Content-Type: application/xml</td>
</tr>
<tr>
<td>Accept: application/json</td>
<td>Accept: application/json</td>
</tr>
<tr>
<td>Host: &lt;instance&gt;.service-now.com</td>
<td>Host: &lt;instance&gt;.service-now.com</td>
</tr>
<tr>
<td>Connection: close</td>
<td>Connection: close</td>
</tr>
<tr>
<td>Content-Length: 91</td>
<td>Content-Length: 91</td>
</tr>
<tr>
<td>[</td>
<td>&lt;request&gt;&lt;entry&gt;</td>
</tr>
<tr>
<td>{ &quot;name&quot;: &quot;user0&quot;,</td>
<td>&lt;name&gt;user0&lt;/name&gt;</td>
</tr>
<tr>
<td>&quot;id&quot;: 1234</td>
<td>&lt;id&gt;1234&lt;/id&gt;</td>
</tr>
<tr>
<td>},</td>
<td>&lt;/entry&gt;</td>
</tr>
<tr>
<td>{</td>
<td>&lt;entry&gt;</td>
</tr>
<tr>
<td>&quot;name&quot;: &quot;user1&quot;,</td>
<td>&lt;name&gt;user1&lt;/name&gt;</td>
</tr>
<tr>
<td>&quot;id&quot;: 5678</td>
<td>&lt;id&gt;5678&lt;/id&gt;</td>
</tr>
<tr>
<td>}</td>
<td>&lt;/entry&gt;</td>
</tr>
<tr>
<td>]</td>
<td>&lt;/request&gt;</td>
</tr>
</tbody>
</table>
Responses

Both requests specify application/json as the Accept header value. This causes either response to use JSON formatting, even if the request content type is XML.

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 04 Aug 2015 15:20:44 GMT
Server: ServiceNow
Connection: close
Set-Cookie: BIGipServerpool_<Instance>=880838154.47166.0000; path=/

{"result":{"id":1234,"id1":5678,"name":"user0","name1":"user1"}}

Scripted REST API example - streaming vs object serialization

These examples demonstrate how to send a JSON response using streaming and using default object serialization.

Streaming vs object serialization

When sending a response, you can send a response as a stream or serialize an object. There are advantages and disadvantages to either approach. Pick a technique based on the needs of your integration.

Generally, if the response object is simple, can be represented as XML or JSON, and is a consistent size, use object serialization. If using a format other than XML or JSON, or if the size of the response varies, use streaming.

Streaming the response

Using a streaming responses provides advantages in response time, instance performance, and content flexibility, but adds additional complexity to the script. When using streaming, you are responsible for formatting the response, setting the response status, and setting the Content-Type header. When streaming a response, the requesting user receives a response quickly because the entire response does not need to be created before starting streaming.

This example demonstrates a Scripted REST Resource script that returns an array of incident records using streaming.

```javascript
/**
 * Sample Scripted REST Resource that returns custom JSON objects with properties from Incident GlideRecords
 * This sample uses ServiceNow JavaScript API to query incident records
 * and then iterates over those records to build and stream a custom JSON object that
 * includes some values from the incidents
 */
(function runOperation(/*RESTServiceRequest*/ request, /*RESTServiceResult*/ response) {
  var writer = response.getOutputStream(),
      hdrs = {},
      table = 'incident',
      record_limit = 100,
      gr = new GlideRecord(table);
```
```javascript
// start building response object
writer.writeString("{"results":[");

// iterate over incident records and build JSON representations to be streamed out.
while (gr.next()) {
    var incidentObj = {};
    incidentObj.number = gr.number + '';
    incidentObj.short_description = gr.short_description + '';
    writer.writeString(global.JSON.stringify(incidentObj));

    if (gr.hasNext()) {
        writer.writeString(",");
    }
}

// close the response object
writer.writeString("]"));
})(request, response);
```

A request to this resource returns the following response.

```json
// sample response
/*
HTTP/1.1 200 OK
Content-Type: application/json
Server: ServiceNow

// sample response number of records returned has been truncated for this example

{
 "results": [
 |
 | "number": "INC0011301",
 | "short_description": "lorem ipsum short description 0 my new incident"
 |
 | "number": "INC0011302",
 | "short_description": "lorem ipsum short description 1 my new incident"
 |
 | "number": "INC0011303",
 | "short_description": "lorem ipsum short description 2 my new incident"
 |
 | "number": "INC0011304",
```
Building an object

Using object serialization allows you to take advantage of ServiceNow provided serialization and content negotiation. When serializing an object instead of streaming, the entire object must be created and serialized before the client receives a response. This may delay the response, or require a large amount of system resources if the response object is very large. Object serialization is available only for XML or JSON responses. Responses using a different format must use streaming.

This example returns the same Incident data as the streaming example, but collects all of the response data in an array before sending the response.
* Sample Scripted REST Resource returns an array of custom JSON objects that include 2 incident properties.
* This sample uses ServiceNow JavaScript API to query incident records and then iterates
* over those records building a custom JSON object that includes 2 values from the incident records.
* note that because we are returning a simple JSON object we can rely on built in serialization
* to set the content-type header as well as response status. The 'result_arr' object will not be returned
* until it has been completely built and stored
*/
(function runOperation(/*RESTServiceRequest*/ request, /*RESTServiceResult*/ response) {
  var table = 'incident',
      record_limit = 100,
      result_arr = [],
      gr = new GlideRecord(table);

  gr.setLimit(record_limit);
  gr.query();

  // iterate over incident records and build JSON representations to be streamed out.
  while (gr.next()) {
    var incidentObj = {};
    incidentObj.number = gr.number + '';
    incidentObj.short_description = gr.short_description + '';
    result_arr.push(incidentObj);
  }

  return result_arr;
})(request, response);

A request to this resource returns the following response.

/*
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Server: ServiceNow

// sample response number of records returned has been truncated for this example
{
  "result": [{
    "short_description": "lorem ipsum short description 0 my new incident",
    "number": "INC0011301"
  },
  {
    "short_description": "lorem ipsum short description 1 my new incident",
    "number": "INC0011302"
  },
  {
    "short_description": "lorem ipsum short description 2 my new incident",
    "number": "INC0011303"
}]
*/
Scripted REST API example - streaming file attachments
This example demonstrates how to send an image attachment to a requesting user as a binary stream.

```javascript
/ **
* Sample Scripted REST Resource that returns a stream of
binary representing an attachment
* This sample uses ServiceNow JavaScript API
GlideSysAttachmentInputStream to get an attachment as a stream then
* users WriteStream to stream the response.
*/
(function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
  var hdrs = {},
      attachment_sys_id = '1852fd52471321009db4b5b08b9a71a9';

  hdrs['Content-Type'] = 'image/jpeg';
  response.setStatus(200);
  response.setHeaders(hdrs);

  var writer = response.getOutputStream();
  var attachmentStream = new GlideSysAttachmentInputStream(attachment_sys_id);
  writer.writeStream(attachmentStream);
})(request, response);
```
A request to this resource returns the following response.

```plaintext
// sample response
/*
HTTP/1.1 200 OK
Set-Cookie: glide_session_store=SYSTEM; Expires=Fri, 30-Oct-2015 21:57:00 GMT; Path=/; HttpOnly
Content-Type: image/jpeg
Transfer-Encoding: chunked
Date: Fri, 30 Oct 2015 21:26:59 GMT
Connection: close
Server: ServiceNow

<binary response body excluded from this sample>
*/
```

**SOAP web service**

The SOAP Web Services provided by ServiceNow are WS-I compliant, as outlined in the WS-I Basic Profile 1.0.

**Concepts and Terminology**

- **Provider** - publishes web service for clients to invoke (consume)
- **Consumer** - invokes / consumes published web service
- **Standards**
  - **WSDL**
    - Web Service Description Language
    - XML document describing functions, arguments, data schema, and endpoint (where / how to invoke the service, URL)
    - WSDL only necessary when generating SOAP envelope programmatically
  - **SOAP**
    - Simple Object Access Protocol
    - XML document usually HTTP posted to web service endpoint described in WSDL
    - SOAP:Envelope / SOAP:Header / SOAP:Body
  - **HTTP**
    - Hyper-Text Transfer Protocol
    - POST vs GET - Web Service is POSTed
    - XML vs. Form POST - Web Service is XML

**Web Service Provider**

ServiceNow publishes its underlying table structures and associated data with the following Web Service methods:

- **Direct Web Services**: Use a URL query to request a ServiceNow table's WSDL.
Web Service Import Sets: Use import tables and transform maps to automate Web Service requests to ServiceNow tables.

Scripted Web Services: Use custom JavaScript to execute Web Service requests.

**Note:** SOAP messages are sent with the assumption that the recipient is XML compliant. No encoding is applied to the SOAP message.

**SOAP direct web service API functions**

The standard SOAP API is a set of small, globally defined functions that can be performed on a targeted resource.

The targeted resource (or table) is defined in the URL by the format `https://<instance name>.service-now.com/<table name>.do`.

---

### Table 1066: Data Retrieval API

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getKeys</td>
<td>Query the targeted table by example values and return a comma delimited <code>sys_id</code> list.</td>
</tr>
<tr>
<td>getRecords</td>
<td>Query the targeted table by example values and return all matching records and their fields.</td>
</tr>
<tr>
<td>get</td>
<td>Query a single record from the targeted table by <code>sys_id</code> and return the record and its fields.</td>
</tr>
<tr>
<td>aggregate</td>
<td>Query using and aggregate functions SUM, COUNT, MIN, MAX and AVG. To enable the aggregate functions, activate the <strong>Aggregate Web Service Plugin</strong>.</td>
</tr>
</tbody>
</table>

---

### Table 1067: Data Modification API

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>insert</td>
<td>Creates a new record for the table targeted in the URL.</td>
</tr>
<tr>
<td>insertMultiple</td>
<td>Creates multiple new records for the table targeted in the URL. To enable multiple inserts, activate the <strong>Web Service Insert Multiple Plugin</strong>.</td>
</tr>
<tr>
<td>update</td>
<td>Updates an existing record in the targeted table in the URL, identified by the mandatory <code>sys_id</code> field.</td>
</tr>
<tr>
<td>deleteRecord</td>
<td>Deletes a record from the targeted table by supplying its <code>sys_id</code>.</td>
</tr>
<tr>
<td>deleteMultiple</td>
<td>Delete multiple records from the targeted table by example values.</td>
</tr>
</tbody>
</table>

---

*Data Retrieval API*

Data Retrieval API method summaries and descriptions.
### Table 1068: Data Retrieval API

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getKeys</td>
<td>Query the targeted table by example values and return a comma delimited <code>sys_id</code> list.</td>
</tr>
<tr>
<td>getRecords</td>
<td>Query the targeted table by example values and return all matching records and their fields.</td>
</tr>
<tr>
<td>get</td>
<td>Query a single record from the targeted table by <code>sys_id</code> and return the record and its fields.</td>
</tr>
<tr>
<td>aggregate</td>
<td>Query using and aggregate functions SUM, COUNT MIN, MAX and AVG. To enable the aggregate functions, activate the <a href="http://www.service-now.com/incident">Aggregate Web Service Plugin</a>.</td>
</tr>
</tbody>
</table>

**getKeys**

Query the targeted table by example values and return a comma delimited `sys_id` list.

**Input fields**

Any field value in the targeted table.

**Output fields**

A SOAP response element `sys_id` that contains a comma delimited list of `sys_id` values.

**Sample SOAP messages**

**Sample SOAP request**

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <inc:getKeys>
      <category>hardware</category>
    </inc:getKeys>
  </soapenv:Body>
</soapenv:Envelope>
```

**Sample SOAP response**

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <getKeysResponse>
    </getKeysResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
Language-specific sample messages

For language-specific `getKeys` samples, refer to the following topics:

- **Perl SOAP::Lite**
- **Java Apache Axis2**
- **Python**

**getRecords**

Query the targeted table by example values and return all matching records and their fields.

**Input fields**

Any field value in the targeted table.

**Output fields**

The `getRecordResponse` element may contain one or more `getRecordsResult` elements that encapsulate elements representing the field values of records matching the query.

**Sample SOAP messages**

**Sample SOAP request**

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <inc:getRecords>
      <number>INC0000002</number>
    </inc:getRecords>
  </soapenv:Body>
</soapenv:Envelope>
```

**Sample SOAP request using an encoded query to filter where incident number is INC00000001 or INC00000002**

```xml
  <soapenv:Header/>
  <soapenv:Body>
  </soapenv:Body>
</soapenv:Envelope>
```
Sample SOAP response that contains 1 record

```
  <soapenv:Header/>
  <soapenv:Body>
    <getRecordsResponse>
      <getRecordsResult>
        <caller_id>5137153cc611227c000b9d1bd8cd2007</caller_id>
        <caller_id.email>david.loo@service-now.com</caller_id.email>
        <closed_at/>
        <number>INC0000002</number>
        <opened_at>2009-12-14 23:07:12</opened_at>
        <short_description>Can't get to network file shares</short_description>
      </getRecordsResult>
    </getRecordsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample SOAP response that contains more than 1 record

```
  <soapenv:Header/>
  <soapenv:Body>
    <getRecordsResponse>
      <getRecordsResult>
        <caller_id>5137153cc611227c000bbd1bd8cd2006</caller_id>
        <caller_id.email>rick.berzle@yourcompany.com</caller_id.email>
        <closed_at>2009-12-17 22:55:16</closed_at>
        <number>INC0000009</number>
        <opened_at>2009-12-16 22:50:23</opened_at>
        <short_description>Reset my password</short_description>
      </getRecordsResult>
      <getRecordsResult>
        <caller_id>5137153cc611227c000bbd1bd8cd2005</caller_id>
        <caller_id.email>fred.luddy@yourcompany.com</caller_id.email>
        <closed_at>2009-12-15 22:54:55</closed_at>
        <number>INC0000010</number>
        <opened_at>2009-12-10 22:53:02</opened_at>
      </getRecordsResult>
    </getRecordsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
Language-specific sample messages

For language-specific getRecords samples, refer to the following topics:

- Perl SOAP::Lite
- Java Apache Axis2
- Python

get

Query a single record from the targeted table by sys_id and return the record and its fields.

Input fields

An element <sys_id> identifying the sys_id of the record to be retrieved.

Output fields

A getResponse element encapsulating all field values for the record retrieved.

Sample SOAP messages

Sample SOAP request

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <inc:get>
      <sys_id>46e18c0fa9fe19810066a0083f76bd56</sys_id>
    </inc:get>
  </soapenv:Body>
</soapenv:Envelope>
```

The resulting response of a get function call looks like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getResponse xmlns="http://www.service-now.com/incident">
      <active>1</active>
    </getResponse>
  </soap:Body>
</soap:Envelope>
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse>
      <number>INC10055</number>
      <opened_at>2007-09-18 00:32:09</opened_at>
      <opened_by>46bac3d6a9fe1981005f299d979b8869</opened_by>
      <priority>0</priority>
      <reassignment_count>0</reassignment_count>
      <severity>0</severity>
      <due_date>2007-10-28 13:29:45</due_date>
      <impact>3</impact>
      <incident_state>1</incident_state>
      <knowledge>0</knowledge>
      <location>1081761cc611227501d063fd3475a2de</location>
      <made_sla>1</made_sla>
      <notify>1</notify>
      <number>INC10055</number>
      <opened_at>2007-09-18 00:32:09</opened_at>
      <opened_by>46bac3d6a9fe1981005f299d979b8869</opened_by>
      <priority>0</priority>
      <reassignment_count>0</reassignment_count>
      <severity>0</severity>
    </getResponse>
  </soap:Body>
</soap:Envelope>

Language-specific sample messages

For language-specific get samples, refer to the following topics:

Perl SOAP::Lite
Java Apache Axis2
Python

aggregate
Query a table using an aggregate function including SUM, COUNT, MIN, MAX, and AVG.

**Note:** Functionality described here requires the Aggregate Web Service plugin.

Input fields

Any element of the target table. In addition, one or more of the aggregate functions (SUM, COUNT, MIN, MAX, and AVG).

A GROUP BY and a HAVING clause may also be added.

Output fields

An aggregateResponse element encapsulating all field values for the record retrieved.
Sample SOAP messages

Sample SOAP request using COUNT aggregate function.

```xml
<?xml version="1.0" encoding="UTF-8"?>
encoding/"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
soap/envelope/"
 xmlns:m="http://www.service-now.com"
 xmlns:tns="http://www.service-now.com/map"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
SOAP-ENV:encodingStyle="http://
schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<aggregate>
<COUNT>number</COUNT>
<active>true</active>
</aggregate>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The resulting response of a COUNT aggregate function call looks like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/
soap/encoding/"
 xmlns:m="http://www.service-now.com"
 xmlns:tns="http://www.service-now.com/map"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
SOAP-ENV:encodingStyle="http://
schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<aggregateResponse>
<aggregateResult>
<avg>2.7200</avg>
</aggregateResult>
</aggregateResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample SOAP request using AVG aggregate function with a GROUP BY clause.

```xml
<?xml version="1.0" encoding="UTF-8"?>
soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/
soap/encoding/"
 xmlns:m="http://www.service-now.com"
 xmlns:tns="http://www.service-now.com/map"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
SOAP-ENV:encodingStyle="http://
schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
The resulting response of a AVG aggregate function call with a GROUP BY clause looks like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  xmlns:m="http://www.service-now.com"
  xmlns:tns="http://www.service-now.com/map"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <aggregateResponse>
      <aggregateResult>
        <avg>1.0000</avg>
        <category>database</category>
      </aggregateResult>
      <aggregateResult>
        <avg>3.0000</avg>
        <category>hardware</category>
      </aggregateResult>
      <aggregateResult>
        <avg>3.0000</avg>
        <category>inquiry</category>
      </aggregateResult>
      <aggregateResult>
        <avg>2.0000</avg>
        <category>network</category>
      </aggregateResult>
      <aggregateResult>
        <avg>2.6923</avg>
        <category>software</category>
      </aggregateResult>
    </aggregateResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Sample SOAP request using an encoded query to filter the aggregate:

```xml
<?xml version="1.0" encoding="UTF-8"?>
xmlns:m="http://www.service-now.com"
xmlns:tns="http://www.service-now.com/map"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <aggregate>
      <COUNT>number</COUNT>
      <active>true</active>
      <__encoded_query>number=INC0000001^ORnumber=INC0000002</__encoded_query>
    </aggregate>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample aggregate request using HAVING to narrow the results.

HAVING takes four fields. Each field is delimited by "^": the aggregate type, the field of the aggregate, the operation type, and the value to compare.

More than one HAVING can be added to the request, so you can use HAVING expressions, but there is no support for OR.

```xml
<?xml version="1.0" encoding="UTF-8"?>
xmlns:m="http://www.service-now.com"
xmlns:tns="http://www.service-now.com/map"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <aggregate>
      <COUNT>sys_id</COUNT>
      <GROUP_BY>internal_type</GROUP_BY>
      <HAVING>COUNT^*^>^10</HAVING>
      <HAVING>COUNT^*^<^20</HAVING>
      <COUNT>sys_id</COUNT>
      <active>true</active>
    </aggregate>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Table 1069: Data Modification API

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>insert</td>
<td>Creates a new record for the table targeted in the URL.</td>
</tr>
<tr>
<td>insertMultiple</td>
<td>Creates multiple new records for the table targeted in the URL. To enable multiple inserts, activate the Web Service Insert Multiple Plugin.</td>
</tr>
<tr>
<td>update</td>
<td>Updates an existing record in the targeted table in the URL, identified by the mandatory sys_id field.</td>
</tr>
<tr>
<td>deleteRecord</td>
<td>Deletes a record from the targeted table by supplying its sys_id.</td>
</tr>
<tr>
<td>deleteMultiple</td>
<td>Delete multiple records from the targeted table by example values.</td>
</tr>
</tbody>
</table>

**Insert**

Creates a new record for the table targeted in the URL.

**Input fields**

All fields from the targeted table, excluding system fields. Fields configured as mandatory in the System Dictionary are reflected in the WSDL with the attribute minOccurs=1.

**Output fields**

Table 1070: Insert method output fields

<table>
<thead>
<tr>
<th>Table type</th>
<th>Output fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>The sys_id field and the display value of the target table (table) are returned.</td>
</tr>
</tbody>
</table>
| Import set     | The sys_id of the import set row, the name of the transformed target table (table), the display_name for the transformed target table, the display_value of the transformed target row, and a status field, which can contain inserted, updated, or error.  

There can be an optional status_message field or an error_message field value when status=error.  
When an insert did not cause a target row to be transformed (skipped because a key value is not specified), the sys_id field will contain the sys_id of the import set row, rather than the targeted transform table. |
### Sample SOAP messages for a regular table

The following example shows an insert that specifies the short description only:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:m="http://www.service-now.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
        <insert xmlns="http://www.service-now.com">
            <short_description xsi:type="xsd:string">This is a test</short_description>
        </insert>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The resulting response looks like this:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:m="http://www.service-now.com"
    xmlns:tns="http://www.service-now.com/incident"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SOAP-ENV:Body>
        <insertResponse xmlns="http://www.service-now.com">
            <sys_id>6b06494fc611227d00b5f87caf618831</sys_id>
        </insertResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Language-specific sample messages

For language-specific `insert` samples, refer to the following topics:
insertMultiple
Creates multiple new records for the table targeted in the URL.

**Input fields**

The `insertMultiple` element may contain 1 or more record tags that contains all fields from the targeted table, excluding system fields. Limit the number of records inserted in a single operation to no more than 200. You can gradually increase this number with subsequent exports if the increase does not negatively impact instance performance.

**Output fields**

The `insertMultipleResponse` tag is followed by 1 or more record tags that contains:

**Table 1071: Insert method output fields**

<table>
<thead>
<tr>
<th>Table type</th>
<th>Output fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>The <code>sys_id</code> field and the display value of the target table (<code>table</code>) are returned.</td>
</tr>
<tr>
<td>Import set</td>
<td>The <code>sys_id</code> of the import set row, the name of the transformed target table (<code>table</code>), the <code>display_name</code> for the transformed target table, the <code>display_value</code> of the transformed target row, and a <code>status</code> field, which can contain <code>inserted</code>, <code>updated</code>, or <code>error</code>. There can be an optional <code>status_message</code> field or an <code>error_message</code> field value when <code>status=error</code>. When an insert did not cause a target row to be transformed (skipped because a key value is not specified), the <code>sys_id</code> field will contain the <code>sys_id</code> of the import set row, rather than the targeted transform table.</td>
</tr>
<tr>
<td>Import set with multiple transforms</td>
<td>The response from this type of insert will contain multiple sets of fields from the regular import set table insert wrapped in a <code>multiInsertResponse</code> parent element. Each set will contain a <code>map</code> field, showing which transform map created the response.</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved. 3267
Sample SOAP messages for a regular table

The following example shows an insert that specifies the short description only:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:inc="http://www.service-now.com/incident">
  <soapenv:Header/>
  <soapenv:Body>
    <inc:insertMultiple>
      <record>
        <short_description>this is test 1</short_description>
      </record>
      <record>
        <short_description>this is test 2</short_description>
      </record>
      <record>
        <short_description>this is test 3</short_description>
      </record>
    </inc:insertMultiple>
  </soapenv:Body>
</soapenv:Envelope>
```

The resulting response looks like this:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:inc="http://www.service-now.com/incident">
  <soapenv:Header/>
  <soapenv:Body>
    <insertMultipleResponse>
      <insertResponse>
        <sys_id>168160ad4a36231200a89091281dc803</sys_id>
        <number>INC0055180</number>
      </insertResponse>
      <insertResponse>
        <sys_id>1681622e4a36231200a8909115e5c388</sys_id>
        <number>INC0055181</number>
      </insertResponse>
      <insertResponse>
        <sys_id>1681626e4a36231200a89091fa3c0aa8</sys_id>
        <number>INC0055182</number>
      </insertResponse>
    </insertMultipleResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
Sample SOAP messages for an import set table

The following example shows an insert that specifies the short description only:

```
  <soapenv:Header/>
  <soapenv:Body>
    <imp:insertMultiple>:</imp:insertMultiple>
      <imp:record>
        <imp:message>one</imp:message>
        <imp:uuid>a</imp:uuid>
      </imp:record>
      <imp:record>
        <imp:message>two</imp:message>
        <imp:uuid>b</imp:uuid>
      </imp:record>
      <imp:record>
        <imp:message>three</imp:message>
        <imp:uuid>c</imp:uuid>
      </imp:record>
    </imp:insertMultiple>
  </soapenv:Body>
</soapenv:Envelope>
```

The resulting response looks like this:

```
  <soapenv:Header/>
  <soapenv:Body>
    <insertMultipleResponse>
      <insertResponse>
        <sys_id>1296b3a0a0b5b73e966fbf9b7acde</sys_id>
        <table>incident</table>
        <display_name>number</display_name>
        <display_value>INC0010033</display_value>
        <status>ignored</status>
        <status_message>No field values changed</status_message>
      </insertResponse>
      <insertResponse>
        <sys_id>1296b48e0a0b5b62513bb5974a7d96</sys_id>
        <table>incident</table>
        <display_name>number</display_name>
        <display_value>INC0010034</display_value>
        <status>ignored</status>
        <status_message>No field values changed</status_message>
      </insertResponse>
      <insertResponse>
        <sys_id>1296b58b0a0b5b468f534659538b9a</sys_id>
        <table>incident</table>
        <display_name>number</display_name>
        <display_value>INC0010035</display_value>
        <status>ignored</status>
        <status_message>No field values changed</status_message>
      </insertResponse>
    </insertMultipleResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
update

Updates an existing record in the targeted table in the URL, identified by the mandatory **sys_id** field.

**Input fields**

All fields from the targeted table, excluding system fields, which will be used for updating the existing record. The **sys_id** field is used to locate the existing record.

**Output fields**

Returns the **sys_id** of the record that was updated.

**Sample SOAP messages**

**Sample SOAP request**

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <inc:update>
      <sys_id>46e18c0fa9fe19810066a0083f76bd56</sys_id>
      <short_description>this is updated</short_description>
    </inc:update>
  </soapenv:Body>
</soapenv:Envelope>
```

**Sample SOAP response**

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <updateResponse>
      <sys_id>46e18c0fa9fe19810066a0083f76bd56</sys_id>
    </updateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
Language-specific sample messages

For language-specific update samples, refer to the following topics:

Perl SOAP::Lite
Java Apache Axis2
Microsoft .NET
Python

deleteRecord
Delete a record from the targeted table by supplying its sys_id.

Input fields

An element <sys_id> identifying the sys_id of the record to be retrieved.

Output fields

A <count> element within the deleteRecordResponse parent element indicating the number of records deleted, this will always equal to "1" for deleteRecord.

Sample SOAP messages

Sample SOAP request

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <inc:deleteRecord>
      <sys_id>46e18c0fa9fe19810066a0083f76bd56</sys_id>
    </inc:deleteRecord>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample SOAP response

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <deleteRecordResponse>
      <count>1</count>
    </deleteRecordResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
Language-specific sample messages

For language-specific deleteRecord samples, refer to the following topics:

Perl SOAP::Lite
Java Apache Axis2
Microsoft .NET
Python

deleteMultiple
Delete multiple records from the targeted table by example values.

Input fields

All fields from the targeted table, including system fields, are used in query-by-example (QBE) fashion to locate records to be deleted. Query example fields can have special prefixes to constrain the search function.

Output fields

A <count> element within the deleteRecordResponse parent element indicating the number of records deleted.

Sample SOAP messages

Sample SOAP request

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <inc:deleteMultiple>
      <category>hardware</category>
    </inc:deleteMultiple>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample SOAP response

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <deleteMultipleResponse>
      <count>6</count>
    </deleteMultipleResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
Language-specific sample messages

For language-specific deleteRecord samples, refer to the following topics:

Perl SOAP::Lite
Java Apache Axis2
Microsoft .NET
Python

WSDL

All ServiceNow tables and import sets dynamically generate WSDL XML documents that describe its table schema and available operations.

You can get a WSDL format by issuing a URL targeting a ServiceNow table with the WSDL parameter, for example:

https://myinstance.service-now.com/incident.do?WSDL

All dynamically generated and served ServiceNow WSDL accessible via HTTP is available for use under the terms defined in the Open Source Initiative OSI - Apache License, Version 2.0 license agreement.

Apache License, Version 2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that
is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.


Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Direct web services

A direct web service is available for any table in the system provided the correct access control is setup.

The supported format of the incoming message is document style literal XML SOAP documents (Document/Literal). To retrieve the direct web service WSDL description and XML schema, point to the relative URL of <tablename>.do?WSDL. For example, to retrieve the WSDL for the Incident table on the online demo system, use the following URL:

https://<instance name>.service-now.com/incident.do?WSDL

Using forms to limit or extend the query response

On occasion, there is a need to limit the number of field values being returned from a SOAP query (get or getRecords) invocation.

Specifying a form view has the effects of:

1. limiting the response elements to contain only the fields on the view
2. specifying reference record field values from referenced fields eg. caller_id.email, this will cause the value of the caller’s email to be returned in the SOAP response

To enable form views for SOAP queries, you may configure the property com.glide.soap.view to be the name of the view you wish to use for all SOAP query response (the default is soap_response). You may also specify the view name as a URL parameter sysparm_view=<view name> when making the SOAP call, for example:

https://<instance name>.service-now.com/incident.do?SOAP&sysparm_view=ess

If a specified view name does not exist, the default behavior is to respond with all fields.

Direct web services extended query parameters

The following parameters, when specified as elements of input parameters to SOAP query functions such as get, getKeys, and getRecords, has the additional behavior of filtering and modifying the results that are returned.

Note: Extended query element names are preceded by two underscore characters.
Table 1072:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>__encoded_query</td>
<td>Specify an encoded query string to be used in filtering the returned results. The encoded query string format is similar to the value that may be specified in <code>asysparm_query</code> URL parameter. You may refer to the encoded query building example in the RSS feed generator examples.</td>
<td><code>&lt;__encoded_query&gt;active=true^category='hardware'&lt;/__encoded_query&gt;</code></td>
</tr>
<tr>
<td>__order_by</td>
<td>Instruct the returned results to be ordered by the specified field</td>
<td><code>&lt;__order_by&gt;prioriy&lt;/__order_by&gt;</code></td>
</tr>
<tr>
<td>__order_by_desc</td>
<td>Instruct the returned results to be ordered by the specified field, in descending order</td>
<td><code>&lt;__order_by_desc&gt;opened_date&lt;/__order_by_desc&gt;</code></td>
</tr>
<tr>
<td>__exclude_columns</td>
<td>Specify a list of comma delimited field names to exclude from the result set</td>
<td><code>&lt;__exclude_columns&gt;sys_created_on, __exclude_columns&gt;</code></td>
</tr>
<tr>
<td>__limit</td>
<td>Limit the number of records that are returned</td>
<td><code>&lt;__limit&gt;100&lt;/__limit&gt;</code></td>
</tr>
<tr>
<td>__first_row</td>
<td>Instruct the results to be offset by this number of records from the beginning of the set. When used with __last_row has the effect of querying for a window of results. The results are inclusive of the first row number.</td>
<td><code>&lt;__first_row&gt;250&lt;/__first_row&gt;</code></td>
</tr>
<tr>
<td>__last_row</td>
<td>Instruct the results to be limited by this number of records from the beginning of the set, or the __start_row value when specified. When used with __first_row has the effect of querying for a window of results. The results are less than the last row number, and does not include the last row.</td>
<td><code>&lt;__last_row&gt;500&lt;/__last_row&gt;</code></td>
</tr>
<tr>
<td>__use_view</td>
<td>Specify a Form view by name, to be used for limiting and expanding the results returned. When the form view contains deep referenced fields eg. caller_id.email, this field will be returned in the result as well</td>
<td><code>&lt;__use_view&gt;soap_view&lt;/__use_view&gt;</code></td>
</tr>
</tbody>
</table>
Return display value for reference variables

When you query a record using a get or getRecords function the instance returns all fields associated with that record. The fields are often reference fields that contain a sys_id for a record on another table.

Use one of these options if you want the display value for the field to be returned instead of the sys_id:

1. Add the property glide.soap.return_displayValue to your system properties, and every SOAP request will return a display value for a reference field.

2. Add the parameter displayvalue=true to your SOAP request URL, and SOAP requests with that parameter will return a display value for a reference field as a string, instead of the sys_id. The SOAP URL would look as follows: https://<instance name>.service-now.com/incident.do?displayvalue=true&SOAP

3. Add the parameter displayvalue=all to your SOAP request URL, and SOAP requests with that parameter will return a display value for a reference field, in addition to the sys_id. The response element name for the display value field will be prefixed with dv_ such as dv_caller_id.

Retrieving journal entries using direct web services

To get the contents of a journal field, make a second soap request against the sys_journal_field table to pull the appropriate journal records back for the record in question.

The URL for the WSDL would be in the following format

https://instance-name.service-now.com/sys_journal_field.do?WSDL

To retrieve the journal entries, you will first need to query the incident for its sys_id value and then supply it as the element_id value in a getRecords call. To sepcify records only for the "comments" field, specify the value "comments" for the element field. For example, a SOAP request would look like the following.

   >
   <soapenv:Header />
   <soapenv:Body>
   <sys:getRecords>
   <element>comments</element>
   <element_id>9d385017c611228701d22104cc95c371</element_id>
   </sys:getRecords>
   </soapenv:Body>
   </soapenv:Envelope>

Retrieving choice fields using direct web services

To retrieve or set choice fields, use the choice Value not the Label.

For example, if you want to retrieve a list of all closed incidents use the numerical value for Closed, which is 7 by default.

<state>7</state>

To see a list of choice values:

1. Navigate to the form containing the choice field. For example, navigate to Incident > Open and select an incident.

2. Right-click the choice value field and select Configure Dictionary (Personalize Dictionary in versions prior to Fuji). For example, configure the dictionary for the State field.

3. From the Choices related list, note the value for the label you want to query. For example, note that the Closed choice has a value of 7.
Persisting an HTTP session across all SOAP calls

In circumstances when a SOAP client makes many calls in a short amount of time, you may want to re-use a single HTTP session for all SOAP calls.

Each SOAP call creates a new user session that persists until it expires. To create a single user session and re-use it for all inbound SOAP calls, develop your SOAP client following these guidelines:

- Use a module like HTTP::Cookies to create a "cookie jar".
- Save the cookies returned by ServiceNow after each request (handled automatically by HTTP::Cookies).
- Re-send the cookies in the cookie jar with each subsequent request.

**Note:** If you have enabled the session rotation high security setting, it will immediately invalidate the JSESSIONID returned from the server with the first response header. The second response includes a new JSESSIONID.
In Perl, you can automatically save and send cookies with the following code:

```perl
use HTTP::Cookies;

# We want to store and re-send cookies
my $cookies = HTTP::Cookies->new(ignore_discard => 1);

my $soap = SOAP::Lite

# Set the cookie jar
$soap->transport->cookie_jar($cookies);
```

**Compatibility for clients generated from WSDL**

Review these guidelines for service namespaces.

### Specifying a Unique Namespace for each Table

The property `glide.wsdl.definition.use_unique_namespace` ensures that each table's Direct Web Service WSDL has a unique targetNamespace attribute. This property is true by default, which requires a table's Direct Web Service WSDL to use a targetNamespace value of `http://www.service-now.com/<table name>`. When false (or when the property is not present), all tables use the same targetNamespace value of `http://www.service-now.com`. Since all tables also share the same operation names, a Web Service client attempting to consume more than one ServiceNow Web Service would be unable to differentiate between requests between multiple tables. Using a unique targetNamespace value allows Web Services clients to distinguish requests between multiple tables.

For example, the Direct Web Service WSDL for the incident table uses this targetNamespace value.

```xml
<wSDL:definitions xmlns:xsd= "http://www.w3.org/2001/XMLSchema"
    xmlns:soap = "http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:soapenc = "http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/">
    <wsdl:types>
        <xsd:schema elementFormDefault = "unqualified"
            targetNamespace = "http://www.service-now.com/incident"
        >
        </xsd:schema>
    </wsdl:types>
</wSDL:definitions>
```

### Setting Namespace Requirements

ServiceNow's WSDL schema by default declares an attribute of `elementFormDefault="unqualified"`. This attribute indicates whether or not locally declared elements must be qualified by the target namespace in an instance document. If the value of this attribute is 'unqualified', then locally declared elements should not be qualified by the target namespace. If the value of this attribute is 'qualified', then locally declared elements must be qualified by the target namespace.

However, this is incompatible with the way clients generated from WSDL (.NET, Axis2, webMethods, etc.) process the embedded schema, it removes the schema namespace as a result, making the web service response unparseable.

To overcome this compatibility issue, a boolean property called `glide.wsdl.schema.UnqualifiedElementFormDefault` is introduced. This property has the value of `true` by default, setting it to `false` will make clients generated from WSDL able to parse the return value of the web service invocation. You can modify this property using the Web Services properties page at **System Properties > Web Services**.
Allowing Duplicate Service Names

By default, service names from dynamically generated WSDL are unique and have the following format:

\[\text{ServiceNow}_<\text{table name}>\]

To allow duplicate service names, administrators can set the glide.wsdl.unique_service_name property to \textit{false}. Create the property if it does not exist.

**Direct web services Perl example - querying all incidents 5 records at a time**

This example query queries all incidents, orders by the number field, and retrieves the first 5 records.

You can easily extend this example to retrieve a set of predefined records, \textit{n} number of records each query, simulating a \textit{windowed} querying client. Using a \textit{windowed} query mechanisms overcomes the default limitation of only getting a maximum of 250 records per query.

```perl
#!/usr/bin/perl -w
use SOAP::Lite; # basic auth using the ITIL user
sub SOAP::Transport::HTTP::Client::get_basic_credentials { return 'itil' => 'itil'; }
# specify the endpoint to connect
my $soap = SOAP::Lite
  -> proxy('https://<instance name>.service-now.com/incident.do?SOAP');
my $method = SOAP::Data->name('getRecords')
  ->attr({xmlns => 'http://www.service-now.com/'});
# get all incidents with a window of 5, starting at row 0, and less than row 5 (total of 5 records)
my @params = ( SOAP::Data->name(__first_row => '0') );
push(@params, SOAP::Data->name(__last_row => '4') );
# the last row number can also be taken as the 'limit' offset by the starting first row
# order by the number field
push(@params, SOAP::Data->name(__order_by => 'number') );
my $result = $soap->call($method => @params);
print_results($result);

sub print_results { my ($result) = @_; my %keyHash = %{ $result->body->{'getRecordsResponse'} }; my $i = 0;
my $size = @{$keyHash{'getRecordsResult'}};
for ($i=0; $i<$size; $i++) {
    my %record = %{$keyHash{'getRecordsResult'}[$i]};
    print "------------------------------ $i ----------------------------
";
    foreach my $kk (keys %record) {
        # print only the number of the incident
        if ($kk eq "number") {
            print "$kk=$record{$kk}\n";
        }
    }
}
```

© 2017 ServiceNow. All rights reserved. 3281
SOAP web service import sets

Web Service Import Sets compliment Direct Web Services and Scripted Web Services to provide a web service interface to Import Sets tables.

This type of web service will transform the incoming data synchronously based on the associated Transform Maps by default.

SOAP web services security

ServiceNow enforces web service security using a combination of basic authentication challenge/response over the HTTP protocol and system level access control using contextual security.

Administrators can control what system resources web services users can access by granting them one of the SOAP roles.

**SOAP roles**

To use SOAP web services, you must have the appropriate role for the operation you want to perform.

Additionally, you must have any other roles required to access the target tables.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>soap</td>
<td>Can perform all SOAP operations.</td>
</tr>
<tr>
<td>soap_create</td>
<td>Can insert new records.</td>
</tr>
<tr>
<td>soap_delete</td>
<td>Can delete existing records.</td>
</tr>
<tr>
<td>soap_ecc</td>
<td>Can query, insert, and delete records on the Queues [ecc_queue] table.</td>
</tr>
<tr>
<td>soap_query</td>
<td>Can query record information.</td>
</tr>
<tr>
<td>soap_query_update</td>
<td>Can query record information and update records.</td>
</tr>
<tr>
<td>soap_script</td>
<td>Can run scripts that specify a .do endpoint. This role is required for running Scripted Web Services.</td>
</tr>
<tr>
<td>soap_update</td>
<td>Can update records.</td>
</tr>
<tr>
<td>import_admin</td>
<td>Can manage all aspects of Import Sets and imports. Required for access to the sys_import_set_row table.</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>import_transformer</td>
<td>Can manage Import Set Transform Maps and run transforms. Required for access to the sys_import_set_row table.</td>
</tr>
</tbody>
</table>

Default web services role requirements

By default, a set of processor access control list (ACL) rules require users to have the soap role in order to make WSDL, XSD, and XML Schema requests.

If you want change these role requirements, you can deactivate the ACL rules.

Figure 807: Web service processor ACLs

Configure SOAP security

Administrators can configure web service security for inbound SOAP requests made to the ServiceNow instance.

Role required: admin

You can also set up web service security to use different certificates for different web service clients. By enabling web service security, you can prevent man-in-the-middle attacks.

**Note:** After you configure a WS-security profile or a security policy, validation is performed on all incoming SOAP requests, including from the MID Server or ODBC driver. Disable validation for these types of requests by marking the service accounts as internal integration users.

1. **Upload a certificate to the instance.**
2. **Create a WS-security profile.**
3. **Create a security policy.**
   Security policies define which WS-security profiles are used to evaluate a particular web service request. If no policy is defined, all WS-security profiles are used to evaluate all requests.
4. **Set the value of the property glide.soap.default_security_policy to the name of the new security policy.**

**Basic authentication for web services**

Enforce basic authentication for web services to require a username and password with a web service request.
To enforce basic authentication for the user associated with the instance for each WSDL or SOAP message request, administrators can set the property `glide.basicauth.required` to `true`.

When enabled, each WSDL and SOAP request must contain an "Authorization" header as specified in the Basic Authentication protocol.

Because web services requests are non-interactive, ServiceNow always requires the Authorization header during a request.

**Note:** Basic Authentication refers to local credentials or LDAP authentication, if configured.

Supplying basic authentication information with every request (whether or not it is required) has the added advantage that ServiceNow can associate web service invocations with the user supplied in the basic authentication credentials. For example, when creating an Incident record, the journal fields lists the user ID contained in the basic authentication header instead of the default Guest user.

### Basic authentication code samples

Samples of basic authentication code for several programming languages and versions.

#### Perl and the SOAP::Lite libraries

To supply basic authentication when using Perl and the SOAP::Lite libraries, you can implement the following function:

```perl
sub SOAP::Transport::HTTP::Client::get_basic_credentials {
    return 'user_name' => 'password';
}
```

#### C# .NET VS 2005 or older

When using C# .NET VS 2005 or older, you can take advantage of the Credentials object. For example:

```csharp

service.ServiceNow proxy = new service.ServiceNow();
service.get getService = newservice.get();
getService.getResponse getServiceResponse = new service.getResponse();

try
{
    proxy.Credentials = cred;
    getService.sys_id = "bf522c350a0a140701972dbf876f1610";
    getServiceResponse = proxy.get(getService);
    catch (Exception ex) { }
```

#### C# .NET VS 2008

When using C# .NET VS 2008, you can take advantage of the ClientCredentials object. For example:

```csharp
Demo_Incident.ServiceNowSoapClient client = new Test08WebService.Demo_Incident.ServiceNowSoapClient();
client.ClientCredentials.UserName.UserName = "admin";
client.ClientCredentials.UserName.Password = "admin";
```
Then in your app.config file look for the following and change "None" to "Basic":

```xml
<transport clientCredentialType="None" proxyCredentialType="None" realm="" />
```

**VB .NET**

When using VB .NET taking advantage of the Credentials object would look like the following:

```vb
Sub Main()
    Dim cred As New System.Net.NetworkCredential("user_name", "password")
    Dim proxy As New VB_Democm.incident.ServiceNow
    Dim getIncident As New VB_Democm.incident.get
    Dim getResponse As New VB_Democm.incident.getResponse
    proxy.Credentials = cred
    getIncident.sys_id = "[your sysID here]"
    getResponse = proxy.get(getIncident)
End Sub
```

The resulting response when Basic Authentication is turned on and no credentials are supplied looks like this:

```html
<html>
<head>
<title>Apache Tomcat/5.0.28 - Error report</title>
<style>   <!--
H1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;}
H2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;}
H3 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;}
BODY {font-family:Tahoma,Arial,sans-serif;color:background-color:white;}
B {font-family:Tahoma,Arial,sans-serif;color:background-color:black;}
P {font-family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;}
A {color: black;}
A.name {color: black;}
HR {color: #525D76;}
-->
</style>
</head>
<body>
<h1>HTTP Status 401 -</h1>
<p>HTTP Status 401 -</p>
<p>This request requires HTTP authentication (</p>
</body>
</html>
```
WS-Security
Validate signed web services requests with WS-security.

ServiceNow supports WS-Security 1.1 to validate signed web services requests. Enable WS-Security to:

• Verify SOAP messages originate from a known sender
• Verify SOAP messages have not been altered in transit

Note: ServiceNow does not use WS-Security as an encryption mechanism. ServiceNow relies on the HTTPS protocol to encrypt all communications.

WS-Security is intended to work in conjunction with basic authentication. When ServiceNow receives a SOAP message, it reviews the basic authentication header to determine if the SOAP user has rights to the instance. It reviews the WS-Security header to determine the validity of the incoming message. Requests affected by attacks such as a man-in-the-middle attack have an invalid WS-Security header and are blocked.

Enable WS-Security verification
Administrators can enable Web Services Security (WSS) verification from the Web Services system properties.
Role required: web_service_admin or admin

1. Navigate to System Web Services > Properties.
2. For Require WS-Security header verification for all incoming SOAP requests, select Yes.
3. Click Save.
4. Create a WS-security profile.
5. Update the user record for the MID Server and ODBC driver to mark these users as internal integration users.
6. Download and install the latest MID Server and ODBC driver.
7. To validate SOAP request signatures, upload the remote web service's certificate as a JKS and create the web service's WSS Username Token Profile.

Note: Because ServiceNow's WSS implementation does not verify the CA certificate, you do not need to upload the web service's CA certificate.

Mark service accounts as internal integration users
Allow internal integration communications to bypass the WSS authentication requirement by marking their user accounts as internal integration users.
Role required: admin

Enabling WSS requires authentication for all SOAP requests including internal integration communications such as the MID Server, ODBC Driver, Remote Update Sets, and high availability cloning. SOAP requests for these internal integration communications cannot implement WSS due to technical implications. If your instance uses these SOAP interfaces, you can allow them to bypass the WSS authentication requirement by marking their user accounts as internal integration users.

1. Navigate to User Administration > Users.
2. Select the user account for the MID Server or ODBC Driver.
3. Configure the form to add the Internal Integration User field.
4. Select the **Internal Integration User** check box.
5. Click **Update**.

WS-Security profiles
A WS-security profile determines how ServiceNow authenticates a web services message when WS-security is enabled.

ServiceNow can authenticate web services requests with the following mechanisms:

<table>
<thead>
<tr>
<th>Authentication mechanism</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate verification</td>
<td>ServiceNow authenticates the web services request by verifying the certificate associated with the request. Verifying the request's certificate requires uploading the requestor's certificate and certificate authority.</td>
</tr>
<tr>
<td>User credentials</td>
<td>ServiceNow authenticates the Web Services request by verifying the user credentials associated with the request. ServiceNow can either verify that the request's credentials match an existing ServiceNow user's credentials or that the request's credentials match a username and password provided in the profile record.</td>
</tr>
</tbody>
</table>

Specify the authentication mechanism you want to use when you create a new WS-security profile.

The WS-Security Profiles module lists the WS-security profiles that are currently in effect.

Create a new WS-Security profile
Create a new WS Security profile to define how ServiceNow authenticates a Web Services message when WS-Security is enabled.

Role required: web_service_admin or admin

1. Navigate to **System Web Services > WS Security Profiles**.
2. Click **New**.
3. Fill in the WS-Security Profile form (see table).
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the security profile.</td>
</tr>
<tr>
<td>Type</td>
<td>Select <strong>X509</strong> to verify the request's certificate. Select <strong>Username</strong> to verify the request's user credentials.</td>
</tr>
<tr>
<td>Run as user</td>
<td>Select the ServiceNow user the instance will impersonate if authentication succeeds and the <strong>Bind Session</strong> field is selected. All web services requests will be attributed to this user. For example, if you select the <strong>System Administrator</strong> user then the instance treats all web services operations as being done by the system administrator. Make sure the user you select has appropriate SOAP privileges if you are using the glide.soap.strict_security high security setting. This field is only visible when the type is <strong>X509</strong>.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter the order in which the instance checks security profiles. The instance checks all security profiles when processing an incoming SOAP request. If a request fails any security profile authentication requirement, the instance stops processing additional security profiles and fails the request.</td>
</tr>
<tr>
<td>Bind Session</td>
<td>Select this check box to have the instance impersonate the user listed in the <strong>Run as user</strong> field. You should only set this field for one profile at a time. If multiple profiles have this field selected, ServiceNow impersonates the user from the last successfully authenticated WS-Security profile. If no profile has this field selected, ServiceNow impersonates the user provided with the basic authentication headers or impersonates the default user (Guest).</td>
</tr>
<tr>
<td>X509 Certificate</td>
<td>Select the certificate record containing the certificate for web service requests. ServiceNow only validates the request signature. It automatically trusts the certificate's certificate authority (CA). This field is only visible when the type is <strong>X509</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Profile action</td>
<td>Select how you want the instance to authenticate the user credentials. Select <strong>Authenticate with user</strong> if you want the instance to authenticate the request based on an existing user record. The request’s credentials must match values in an existing user record. Select <strong>Specify user to authenticate</strong> if you want to list a user name and password combination that all web services requests must meet. The request's credentials must match the user name and password you list. This field is only visible when the type is <strong>Username</strong>.</td>
</tr>
<tr>
<td>User field to match UserName</td>
<td>Select the column from the User [sys_user] table containing the value you want match against the request's UserName. For example, if you select <strong>Email</strong> then the request UserName header must contain an email address matching an existing ServiceNow user. This field is only visible when the profile action is <strong>Authenticate with user</strong>.</td>
</tr>
<tr>
<td>User name</td>
<td>Enter the user name that all web services requests must contain. This field is only visible when the profile action is <strong>Username</strong>.</td>
</tr>
<tr>
<td>User password</td>
<td>Enter the password that all web services requests must contain. This field is only visible when the profile action is <strong>Username</strong>.</td>
</tr>
</tbody>
</table>

4. **Click Submit.**

**WSS X.509 Token Profile**

Use the X.509 framework for a WSS X.509 security profile.

An X.509 certificate is used to validate a public key that is used to sign the incoming SOAP message. It specifies a binding between a public key and a set of attributes that includes (at least) the following:

- subject name
Geneva    ServiceNow    ServiceNow Platform

- issuer name
- serial number
- validity interval

Use the **X.509 authentication framework** as defined by the *Web Services Security: SOAP Message Security specification*.

Upload the certificate and reference it in the **X509 Certificate** field. If this is a bound session, select the user to impersonate when the WS-Security authentication succeeds.

![WSS X.509 Security Profile](image)

**Figure 809: WSS X.509 Security Profile**

**WSS UsernameToken Profile**

When specifying the X.509 Token Profile, you can also supply a UsernameToken in the SOAP request.

A UsernameToken is used as a means of identifying the requestor by "username", and optionally using a password (or shared secret, or password equivalent) to authenticate that identity to ServiceNow.

There are two ways to authenticate a UsernameToken:

1. **Authenticate with existing ServiceNow user credentials.**
   
   ![Authenticate with existing ServiceNow user credentials](image)

   **Figure 810: Authenticate with existing ServiceNow user credentials**

   Use the username of the incoming SOAP request to look up a user in ServiceNow by the specified User field to match the UserName value. The password value in the incoming UsernameToken is used to authenticate the request. When the **Bind session** option is selected, the user that authenticates successfully will be used for the session.

2. **Authenticate with specified user credentials.**
Figure 811: Authenticate with specified user credentials

Authenticate using login credentials unrelated to users in the User table. When the Bind session option is selected, the user that is specified in the Run as user field will be used for the session.

Note: The UsernameToken Profile cannot be used independent of the X.509 Token Profile.

Example WS-Security SOAP envelope header

An example of a valid WS-Security SOAP envelope header.

```xml
<SOAP-ENV:Header>
      MIIEgzCCA2ugAwIBAgILAQAAAAABLOZQMtEwDQYJKoZIhvcNAQEFBQAwQDEXMBUGA1UEChMOQ3liZXJ0cnVzdCBJbmMxJTAjBgNVBAMTH1NlcnZpY2UtTm93IFBhcn3uZXIgRGV2ZwcG1lbnQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCvctcRiBz6kGnN9uyhtcSDNS1uCW6FgtnTbTDUvw2nG1NAy9iEvwTpSTG3eELOOFBCuLeYx5281n+cj72v+zCwi/rZcbEPj8oWLyVAOqJTHgrzhdjboVDM/zU8bvAXcw6foCUDFKkc64W7C4yHpBdfW4JT7FBGQ3LEudq80Up+TFETEvRE3jRgy9f92TKD7MN5vKzyhz2xZLOpIN5HJix19juNjMgLwugqTGO4yZSUcItc1jgCyU0+f0NXKgh0QRHeNpwcqWbbJvLBry9ysbo613UAYCQ09hrRnT7VaPvfiueUvulopap04Kei1iL8aMUAEUDtKfi1AbqRIIQ5AgMBAAGjgFFMIIBqTAfBgNVH
SMEGDAwBRJTJLzuOjts557p5VM2taRMACITA7BgNVHR8ENDARYmDcLqAshipodHRwOi8vY3JsLm9tbmblyb29lOmNvbS9Tc0JQ3J1ZGVudGlhCjcmwwHQQYDVR0OBBYEFB+OqlvcdiYmq0enW6mgaVw2p9eAMA8GAUdEBw/wQFMAAMCAQAQdVgDVR0PAQIBAQD
AgTwMBEGCWCAGSAGEIeBAQQEAAwIFoDBJQg3rBgEFPQcBAQ
```

© 2017 ServiceNow. All rights reserved.
SOAP security policies

The Enhanced Web Service Provider - Common plugin adds the SOAP Security Policies module to the System Web Services application.

This module allows administrators to set the following security policies.

- Enable or disable signing SOAP requests when consuming an external web service
- Specify the authentication requirements SOAP requests must meet when communicating over WS-Security.

Activate the Enhanced Web Service Provider - Common plugin
Administrators can activate the Enhanced Web Service Provider - Common plugin to enable unsigned WS-Security requests and specify what authentication requirements SOAP requests have.

Role required: admin

1. Navigate to System Definition > Plugins.
2. Right-click the plugin name on the list and select Activate/Upgrade.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.

3. Optional: If available, select the Load demo data check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

4. Click Activate.

Installed with the Enhanced Web Service Provider - Common plugin

The following components installed with the Enhanced Web Service Provider - Common plugin.

The Enhanced Web Service Provider - Common plugin installs the following components:

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Component Installed</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>Web Services Security Profiles</td>
<td>The plugin adds this module to the System Web Services application.</td>
</tr>
<tr>
<td>System Property</td>
<td>glide.soap.default_security_policy</td>
<td>Specifies the default security policy to use when enforcing Web Services-Security (WSS) for inbound requests.</td>
</tr>
</tbody>
</table>

Certificates required for signed SOAP requests

In order to sign SOAP requests for WS-Security communications, ServiceNow requires the following certificates.

- The X.509 certificate from the requester
- The X.509 CA certificate of the certificate authority who signed the requester’s certificate

Create a new security policy

Administrators can specify which security profiles WS-Security communications must meet by creating a new security policy.

Role required: web_service_admin or admin

2. Click New.
3. Fill out the SOAP Security Policy form (see table).
Table 1077: SOAP Security Policy form fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the security policy. Use this name to set the default security policy with the glide.soap.default_security_policy property.</td>
</tr>
<tr>
<td>Type</td>
<td>Select whether the SOAP security policy applies to inbound or outbound traffic.</td>
</tr>
<tr>
<td>Required to Sign SOAP Request</td>
<td>Select this checkbox to require signed SOAP requests. Clear the checkbox to allow unsigned SOAP requests. When enabled, the instance will produce an error for any SOAP request that does not include a valid signature. When disabled, the instance still verifies any signature included with a SOAP request.</td>
</tr>
<tr>
<td>Authenticate</td>
<td>Select if a SOAP request must authenticate against all security profiles or at least one security profile.</td>
</tr>
<tr>
<td>Security Profiles</td>
<td>Select the security profiles you want to apply to this SOAP security policy. You must select at least one security profile.</td>
</tr>
</tbody>
</table>

4. Click **Submit**.

Specify requirement for signed SOAP requests

Use a SOAP security policy to specify whether the instance requires signed SOAP requests for all inbound SOAP traffic.

Role required: web_service_admin or admin

By default, all inbound SOAP traffic must be signed. Administrators may want to disable this policy and allow unsigned SOAP requests to ServiceNow web services.

1. Navigate to **System Web Services > SOAP Security Policies**.
2. Select the **Default Policy**.
3. Clear the **Required to Sign SOAP Request** check box (selected by default) to disable the requirement.
4. Click **Update**.

SOAP default security policy

Administrators can specify the SOAP security policy an instance uses with the system property glide.soap.default_security_policy.

The glide.soap.default_security_policy system property specifies the name of the SOAP security policy the instance uses when enforcing Web Services-Security (WSS) for inbound requests.
Table 1078: SOAP default security policy settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>Default value</td>
<td>Default Security Policy</td>
</tr>
<tr>
<td>Location</td>
<td>Add to the System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>

Set the SOAP default security policy
Set the SOAP default security policy.
Role required: web_service_admin or admin

1. Navigate to System Web Services > Properties.
2. In the Security Policy to enforce if WS-Security is enabled field, enter the default security policy to use when enforcing WS-security.
3. Click Save.

WS-Security error logging
The glide.processor.debug.SOAPProcessor system property allows error messages about WS-security to be displayed in the transaction log.

The system property glide.processor.debug.SOAPProcessor enables (true) or disables (false) debugging messages for SOAP processing such as certificate and keystore checks.

Table 1079: glide.processor.debug.SOAPProcessor fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Default value</td>
<td>false</td>
</tr>
<tr>
<td>Location</td>
<td>Add to the System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>

WS-Security error messages
An instances produces one of the following error messages when it encounters an issue with a WS-security SOAP message.

Table 1080: WS-security error messages

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Security Policy Selected. Select an Inbound policy for Inbound Requests.</td>
<td>The default policy name is set to an outbound policy. Set the default policy name to an inbound security policy.</td>
</tr>
<tr>
<td>SOAP request not Signed. Policy requires signing.</td>
<td>The SOAP security policy requires signing and the inbound SOAP request is not signed. Either specify a different SOAP security policy or provide the SOAP request with a proper signature.</td>
</tr>
<tr>
<td>Error</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>No profiles to authenticate.</td>
<td>The selected Security policy either does not have any security profiles (X509 or UserNameToken) or the security profiles are inactive. Verify at least one security profile is active.</td>
</tr>
<tr>
<td>Unable to verify signed request.</td>
<td>The inbound SOAP request contains an invalid signature. The SOAP request changed after being signed.</td>
</tr>
<tr>
<td>Failed to extract principal(s) from request.</td>
<td>The SOAP request has either been tampered or was not well formed. ServiceNow cannot extract credentials to authenticate the request.</td>
</tr>
<tr>
<td>Failed to authenticate WS-security, unknown type.</td>
<td>The SOAP request contains an unsupported security profile. Resend the request with a supported security profile type: X509 or UsernameToken.</td>
</tr>
<tr>
<td>Failed to authenticate WS-security.</td>
<td>ServiceNow failed to authenticate against the provided profile credentials. Verify that the SOAP request is using the proper credentials.</td>
</tr>
</tbody>
</table>

**Strict security for web services**

Strict security for web services requires that users meet Contextual Security requirements to access instance resources.

By default, basic authentication for web services only determines whether a user is authorized to access the instance with a SOAP connection. Once authorized, any user can access any table published as a web service.

The system property **Enforce strict security on incoming SOAP requests** changes this behavior and requires that users meet Contextual Security requirements to access instance resources from web services.

With this property enabled, only users that have the proper SOAP role and also meet the Access control rules on page 2498 conditions for a given table and operation can perform that operation from a SOAP connection.

Enforce strict security for inbound SOAP

Strict security for web services requires that users meet Contextual Security requirements to access instance resources.

Role required: admin

**Note:** ServiceNow does not support digital certificates, digital signatures, or other digested token methods in SOAP web service calls.

To enforce strict security for web services connections:

1. Navigate to System Properties > Web Services.
2. Select Yes for Enforce strict security on incoming SOAP requests.
**Mutual authentication for web services**
ServiceNow supports mutual authentication for outbound web services.

*Mutual authentication* is not available for inbound web services.

**Overriding the SOAP endpoint**

The SOAP endpoint address where the SOAP message is posted is consistent with the endpoint of the WSDL.

In some cases, however, the WSDL may reference an incorrect endpoint URL. If this happens, it is necessary to over-ride the generated URL by creating the property `com.glide.soap_address_base_url` to contain the new URL. You may have to add the property manually as this is not an out-of-box property.

For instance, a generated SOAP endpoint may look like this:

```
https://instance.service-now.com/incident.do?SOAP
```

You can specify a property to define the endpoint such that it goes through a proxy by setting the property:

```
com.glide.soap_address_base_url = "https://myproxy.mycompany.com/service-now/"
```

This will result in the endpoint being generated to appear as:

```
https://myproxy.mycompany.com/service-now/incident.do?SOAP
```

**Enabling HTTP compression**

By default, the SOAP request is accepted un-compressed and the result of the request is returned un-compressed.

To enable HTTP compression using [gzip] when sending in your SOAP request, set the following HTTP header:

```
Content-Encoding: gzip
```

To receive the SOAP response compressed using [gzip] send in your SOAP request with the following HTTP header:

```
Accept-Encoding: gzip
```

**Debugging incoming SOAP envelope**

To capture incoming SOAP envelope XML in the system log, add the property `glide.processor.debug.SOAPProcessor` with a value of `true`. 
When enabled, this property adds the incoming SOAP envelope in the **Message** field of the system log (**System Logs > All**). Disable this debugging feature as soon as you are finished so that the log is not overwhelmed with excessive and unnecessary debugging information.

**Preventing empty elements in SOAP messages**

By default, an instance does not omit empty elements, elements with NULL or NIL values, from SOAP messages.

To prevent SOAP responses from containing empty elements, an administrator can create a system property called `glide.soap.omit_null_values` and set the value to **true**. This behavior is compliant with the WSDL as all elements in a SOAP message have a minOccurs=0 attribute and are therefore optional. In addition, this behavior prevents the instance from creating inefficient SOAP messages containing a large number of empty elements.

Set this property to **false** to allow SOAP messages to search for existing fields with empty values. For example, if an administrator wants to search for incidents with an empty **Assigned to** field from a SOAP message, then the SOAP message must be able to send an empty value for this field.

**Note:** Changing the value of this property may cause unintended actions in existing web service integrations. Administrators are strongly encouraged to carefully test the new behavior to ensure that existing integrations process empty elements as expected.

**Insert related records using SOAP**

Support is available for inserting hierarchical data into tables or web service import set tables. The hierarchical data in the Insert API is automatically mapped to related records of the targeted table.

Role required: admin

Create and set the property `glide.web_service.hierarchical` to **true**.

The client of the API can also override this value by invoking the SOAP web service with the URL parameter `hierarchical=true`.

For example, when a related list is created for the incident table called `u_custom_comments`:  

© 2017 ServiceNow. All rights reserved.  3299
Figure 813: Hierarchical Incident

And `u_comment_items` is created as a related list for `u_custom_comments`:
Figure 814: Hierarchical Custom Comments

**WSDL Schema with related records**

When a WSDL for the target Incident table is requested with an additional parameter of hierarchical=true, the WSDL schema for the Insert function will reflect available related records that may participate in the hierarchical data payload.

For example, the insert portion of the schema of the incident table, when requested with hierarchical=true displays its hierarchy as follows:

https://instance.service-now.com/incident.do?WSDL&hierarchical=true
<xsd:element name="insert">
  - <xsd:complexType>
    - <xsd:sequence>
      <xsd:element minOccurs="0" name="active" type="xsd:boolean"/>
      <xsd:element minOccurs="0" name="activity_due" type="xsd:string"/>
      <xsd:element minOccurs="0" name="approval" type="xsd:string"/>
      <xsd:element minOccurs="0" name="approval_history" type="xsd:string"/>
      <xsd:element minOccurs="0" name="approval_set" type="xsd:string"/>
      <xsd:element minOccurs="0" name="assigned_to" type="xsd:string"/>
      <xsd:element minOccurs="0" name="assignment_group" type="xsd:string"/>
      <xsd:element minOccurs="0" name="business_duration" type="xsd:string"/>
      <xsd:element minOccurs="0" name="business_stc" type="xsd:integer"/>
      <xsd:element minOccurs="0" name="calendar_duration" type="xsd:string"/>
      <xsd:element minOccurs="0" name="calendar_stc" type="xsd:integer"/>
      <xsd:element minOccurs="0" name="caller_id" type="xsd:string"/>
      <xsd:element minOccurs="0" name="category" type="xsd:string"/>
      <xsd:element minOccurs="0" name="category_id" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

Figure 815: WSDL Schema

...
The WSDL above shows the incident table having a related table `u_custom_comments` that itself has a related table `u_comment_items`.

**Hierarchical SOAP Message**
When the SOAP message is constructed from the hierarchical web service described by the WSDL and invoked, it will create the incident, `u_custom_comments`, and `u_comment_items` records.

**Endpoint URL**

https://instance.service-now.com/incident.do?SOAP&hierarchical=true
Request

```xml
<soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:inc="http://www.service-now.com/incident">
   <soapenv:Header/>
   <soapenv:Body>
      <inc:insert>
         <short_description>test hierarchical</short_description>
         <u_custom_comments>
            <u_comment>comment 1</u_comment>
            <u_comment_type>travel</u_comment_type>
            <u_comment_items>
               <u_name>name 1</u_name>
               <u_value>value 1</u_value>
            </u_comment_items>
         </u_custom_comments>
      </inc:insert>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```xml
<soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:inc="http://www.service-now.com/incident">
   <soapenv:Header/>
   <soapenv:Body>
      <insertResponse>
         <sys_id>8422ebe7c0a8006e7d23848c2dc8ba47</sys_id>
         <number>INC0010001</number>
      </insertResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

SOAP session management and reporting

A SOAP session is a Glide session established with an instance by any external SOAP client, such as a web services client application, a ServiceNow MID Server, or the ServiceNow ODBC driver.

SOAP sessions are included in the list of user sessions at User Administration > Logged in users.
SOAP sessions are identified by the ?SOAP URLs.

**SOAP session properties**

Certain properties control how SOAP sessions are maintained.
### Table 1081: SOAP session properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.soap.invalidate_session_timeout</td>
<td>The duration, in seconds, that an active session will remain open. After this duration is reached, the instance deactivates the session and reclaims any system resources. If the client sends another request after the timeout duration is reached, the instance will establish a new session. This property accepts values between 5 seconds and 1200 seconds (20 minutes).</td>
</tr>
</tbody>
</table>
|                                              | • Type: integer  
|                                              | • Default value: 60  
|                                              | • Location: Add to the System Properties [sys_properties] table |

**Long-running SOAP request support**

The ServiceNow platform supports long-running SOAP requests by preventing socket timeouts due to inactivity of the network connection while the requests are in process.

This functionality improves the efficiency of the ODBC driver when requesting large numbers of records, doing aggregate queries, or using order by expressions that require sorting.

By default, ServiceNow provides timeout protection for Web Services clients provided by ServiceNow such as the ODBC driver and the MID Server. You can add timeout protection to customer developed Web Services with the parameters described below.

**Timeout Protection**

Web Services clients receive a 307-Temporary Redirect to keep long sessions alive and prevent a timeout due to socket inactivity. A 307-Temporary Redirect causes Web Services clients which support the status code to repeat their last request to the location specified in the HTTP location header. The value of the location header sent by ServiceNow is the same URL that the Web Services client originally specified. The use of 307-Temporary Redirects is WS-I compliant.

A Web Service request that exceeds the timeout limit (glide.soap.request_processing_timeout) can only receive a 307-Temporary Redirect when all of these are met:

- The value of glide.soapprocessor.allow_long_running_threads is true  
- The request includes a redirectSupported=true URL parameter  
- The Request is session-aware (supports HTTP cookies)

If any of these conditions is not met, the Web Service client receives a 408 Request Timeout error.

**Note:** To ensure that applications experience a socket timeout rather than a 408 Request Timeout, set the glide_soap_request_processing_timeout property to a value larger than the shortest socket timeout setting in effect for the connection between the application and the ServiceNow instance (300 seconds for hosted instances).
Properties

The following properties control long-running SOAP processes:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.connection.timeout</td>
<td>Specify the maximum number of milliseconds an outbound HTTP request (such as Web Services) has to finish processing before the connection times out.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 100000 (100 seconds)</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.http.timeout</td>
<td>(Web Service Consumer Plugin) Specifies the maximum number of milliseconds to wait before an outbound transaction times out.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 175000 (175 seconds)</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Properties [sys_properties] table</td>
</tr>
<tr>
<td>glide.soap.request_processing_timeout</td>
<td>Specify the maximum number of seconds an inbound SOAP request has to finish processing before the connection times out. This property computes a default value from the value of the property glide.http.timeout divided by 1000.</td>
</tr>
<tr>
<td></td>
<td>This property accepts values between 5 seconds and 1200 seconds (20 minutes).</td>
</tr>
<tr>
<td></td>
<td>Customers might have network infrastructure (such as proxy servers) in place which implement a shorter timeout. In this case, a socket timeout may occur unless this property is set to a shorter value. In general, you should set this property to a value several seconds less than the shortest socket inactivity timeout in effect anywhere in the network path between the client application and the ServiceNow instance.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 60</td>
</tr>
<tr>
<td></td>
<td>• Location: System Property [sys_properties] table</td>
</tr>
<tr>
<td>glide.soapprocessor.allow_long_running_threads</td>
<td>Enables or disables a 307-Temporary Redirect. The default setting is true (enabled).</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>glide.soapprocessor.max_long_running_threads</td>
<td>Controls the maximum number of long-running SOAP threads which can run at any one time. The default value for this property is determined dynamically based on the number of SOAP semaphores configured. It should not be necessary to change this value.</td>
</tr>
</tbody>
</table>

**Viewing a SOAP session log**

Role required: admin

1. Navigate to User Administration > Logged in users.

![Logged in Users](image)

2. Open an active SOAP session to see the transactions log.

The SOAP session is marked as inactive within 60 seconds of the last transaction.

**AttachmentCreator SOAP web service**

Attaching documents to records in ServiceNow can be achieved by sending a SOAP message targeting the ecc_queue table.

**Important:** The AttachmentCreator SOAP web service is not recommended. Instead, use the REST Attachment API on page 3146 to manage attachments with web services.

Using the AttachmentCreator SOAP web service, you can attach a single document to a message that is a maximum of 5 MB. The following is an example of a URL or target end point: `https://instance_name.service-now.com/ecc_queue.do?SOAP`
### Table 1083: ecc_queue Fields for Attachment Creation

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>The name of the agent sending in the request, this can be any value since it is not used for processing.</td>
<td>AttachmentCreator</td>
</tr>
<tr>
<td>topic</td>
<td>The topic of the queue record, this value must be set to &quot;AttachmentCreator&quot; to trigger the attachment creation</td>
<td>AttachmentCreator</td>
</tr>
<tr>
<td>name</td>
<td>This field must contain a &quot;:&quot; delimited value of the file name, and its content-type</td>
<td>file_name.xls:application/vnd.ms-excel</td>
</tr>
<tr>
<td>source</td>
<td>This field must contain a &quot;:&quot; delimited value of the target table and its sys_id</td>
<td>incident:dd90c5d70a0a0b39000aac5ae704ae8</td>
</tr>
<tr>
<td>payload</td>
<td>This field must contain the Base 64 encoded string representing the object to be attached</td>
<td>the base 64 encoded string</td>
</tr>
</tbody>
</table>

Sending in the values described in the table above will attach an Excel file to the incident table for the record located by the sys_id "dd90c5d70a0a0b39000aac5ae704ae8"

### Security

Like all other HTTP based web services available on the platform, the AttachmentCreator SOAP web service is required to authenticate using basic authentication by default. The user ID that is used for authentication will be subjected to access control in the same way as an interactive user.

To create attachments, the SOAP user must have any roles required to create Attachment [sys_attachment] records, as well as the soap_create role, and any roles required to read and write records on the target table, such as the itil role to add attachments to incident records. By default there is no single role allowing you to add attachments. You can create a role to explicitly allow adding attachments, then assign this role to the SOAP user.

### File type security

You can control what file types users can attach by setting the glide.attachment.extensions and glide.security.file.mime_type.validation properties.

For these properties to apply to the AttachmentCreator web service, the property glide.attachment.enforce_security_validation must be set to true. This property is true by default.
Example SOAP Message

The following is an example of a SOAP message that would take a text file "john1.txt" of mime-type: text/plain and attach it to an Incident with a GUID of: e6eed6950a0a3c59006f32c8e3ff3cf9. Note the payload is the base64 encoding of the file itself.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:ecc="http://www.service-now.com/ecc_queue">
 <soapenv:Header />
 <soapenv:Body>
  <ecc:insert>
   <agent>AttachmentCreator</agent>
   <topic>AttachmentCreator</topic>
   <name>john1.txt:text/plain</name>
   <source>incident:e6eed6950a0a3c59006f32c8e3ff3cf9</source>
   <payload>SSB3b25kZXIgaWYgc2hlIGtub3ducyB3aGF0IHNoZSdzIGRvaW5nIG5vdy4K</payload>
  </ecc:insert>
 </soapenv:Body>
</soapenv:Envelope>
```

Example Node.js Script

The following example Node.js script adds an attachment to an incident record. Run this script from a client computer, not an instance.

```javascript
/**
 * Node.js to ServiceNow attachment upload via SOAP
 * Andrew Venables andrew.venables@servicenow.com
 * July 2014
 * Version 1.0
 *
*/

var soap = require('soap'), // https://github.com/vpulim/node-soap
    mime = require('mime'), // https://github.com/broofa/node-mime
    fs = require('fs');

var WSDL_FILENAME = 'ecc_queue.xml'; // Goto https://instancename.service-now.com/ecc_queue.do?WSDL and save a copy of the WSDL locally for simplicity
var DIRECTORY_CONTAINING_FILES = '/Users/andrew.venables/Documents/Uploads'; // Local path to the directory containing all the files we want to upload
var USERNAME = 'andy.venables'; // An admin user account on the instance
var PASSWORD = 'MY_PASSWORD'; // Password for above account
var TARGET_TABLE = 'incident'; // Target table to attach the files to
var TARGET_SYS_ID = '9d385017c611228701d22104cc95c371'; // Target record / sys_id to attach the files to. OOTB INC0000002

var files_to_upload; // Global that will contain our list of files to be uploaded
var pos = 0; // Global pointer for our position in the files_to_upload list

// Create a SOAP client to use to post to the instance
```
soap.createClient(WSDL_FILENAME, function(err, client) { // Node uses callbacks
  if (err) console.error(err);

  // Set the username and password
  client.setSecurity(new soap.BasicAuthSecurity(USERNAME, PASSWORD));

  // Read all the files in our source directory, will include . and..
  files_to_upload = fs.readdirSync(DIRECTORY_CONTAINING_FILES);

  console.log('Files to upload: ' + files_to_upload.length + '\n');

  // Start iterating through the list of files to upload
  next(client);
});

// Process the next file in the files_to_upload array
// This is a neat way of dealing with Node and its expectation of callbacks
function next(client) {

  // Check we haven't reached the end
  if (pos >= files_to_upload.length) return;

  // Get the next file to upload
  var file_name = files_to_upload[pos];

  // Increment the pointer to the next file
  pos++;

  console.log(pos + '/'+ files_to_upload.length+ ' - Uploading file: ' +
             file_name);

  // A blank file is the end of the list
  if (file_name == '') return;

  // Skip to the next file as this one is invalid
  if (file_name == '.' || file_name == '..' || file_name.indexOf('.') == 0)
    next(client);

  // Get the file type using an module called mime
  var file_type = mime.lookup(file_name);
  console.log('   of type: ' + file_type);

  var file_payload;
  // Load the file into a buffer
  fs.readFile(DIRECTORY_CONTAINING_FILES + '/' + file_name, function(err,
                 the_data) {
    if (err) console.error(err);

    // Encode the buffer to base64
    file_payload = new Buffer(the_data, 'binary').toString('base64');

    // Set the parameters before we call the Web Service
    var parameters = {
      'agent': 'AttachmentCreator',
      'topic': 'AttachmentCreator',
      'name': file_name+':'+file_type,
      'source': TARGET_TABLE+':'+TARGET_SYS_ID,
      'payload': file_payload
    };

    console.log('      sending...')
    // Make the Web Service call, remember node likes callbacks
client.insert(parameters, function(err, result) {
    if (err) console.error(err);
    console.log(result);
    // This file is done, next!
    next(client);
});

Example Perl Script

The following example Perl script will create an attachment to an incident record.

use MIME::Base64;
use strict;
use SOAP::Lite;

# the ServiceNow instance
my $SNC_HOST = "https://instance_name.service-now.com";
my $base64;
my $buf;

# upload and attach a file on the local disk, base 64 encode it into a string first
open(FILE, "~/Users/davidloo/Desktop/test_files/number_test.xls") or die "$!";
binmode FILE; #preserves file formatting on Windows
while (read(FILE, $buf, 60*57)) {
    $base64 .= encode_base64($buf);
}

# call the sub routine to attach
attach_incident();

sub attach_incident {
    # target the ecc_queue table
    my $soap = SOAP::Lite-&gt;proxy("$SNC_HOST/ecc_queue.do?SOAP");
    $soap-&gt;{transport}-&gt;{proxy}-&gt;{ssl_opts}-&gt;{verify_hostname} = 0;
    my $method = SOAP::Data-&gt;name('insert')-&gt;attr({xmlns =&gt; 'http://www.service-now.com/'});

    # set the ecc_queue parameters
    my @params = (SOAP::Data-&gt;name(agent =&gt; 'AttachmentCreator'));
    push(@params, SOAP::Data-&gt;name(topic =&gt; 'AttachmentCreator'));

    # identify the file name and its mime type
    push(@params, SOAP::Data-&gt;name(name =&gt; 'number_test.xls:application/vnd.ms-excel'));

    # attach to the incident, specifying its sys_id
    push(@params, SOAP::Data-&gt;name(source =&gt; 'incident:dd90c5d70a0b39000aac5ae704ae8'));

    # set the payload to be the base 64 encoded string representation of the file
    push(@params, SOAP::Data-&gt;name(payload =&gt; $base64));

    # invoke the web service
    my $result = $soap-&gt;call($method =&gt; @params);
Perl API

The Perl API provides a library of Perl classes and sub routines for programmatic access to the platform and its applications. The API utilizes the SOAP web service interface of the platform.

For more information on the Perl programming language, see [www.perl.org](http://www.perl.org).

Perl API system requirements
This Perl API requires a minimum Perl version and several modules.

The ServiceNow Perl API requires Perl 5.8 (or later) with the following modules installed:

- SOAP::Lite (prerequisites [http://soaplite.com/prereqs.html](http://soaplite.com/prereqs.html)) 0.71 or later
- Crypt::SSLeay
- IO::Socket::SSL
- File::Basename
- MIME::Types
- MIME::Type
- MIME::Base64

**Attention:** Certain versions of the Perl API may not be compatible with the ServiceNow software and may cause 443 connection errors. ServiceNow recommends against using versions 5.14.2, 5.13.6, and 5.16.1 of Perl.

**Install the Perl API**
Download and install the Perl API.

1. Download the Perl API.
The Perl API is available for use under the terms defined in the Open Source Initiative OSI - Apache License, Version 2.0 license agreement.

Apache License, Version 2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding...
communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.


Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such
Derivative Works as a whole, provided Your use, reproduction, and
distribution of the Work otherwise complies with the conditions stated
in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally
submitted for inclusion in the Work by You to the Licensor shall be
under the terms and conditions of this License, without any additional
terms or conditions. Notwithstanding the above, nothing herein shall
supersede or modify the terms of any separate license agreement you may
have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names,
trademarks, service marks, or product names of the Licensor, except as
required for reasonable and customary use in describing the origin of
the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor
provides the Work (and each Contributor provides its Contributions) on
an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR
A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any risks
associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including
negligence), contract, or otherwise, unless required by applicable
law (such as deliberate and grossly negligent acts) or agreed to in
writing, shall any Contributor be liable to You for damages, including
any direct, indirect, special, incidental, or consequential damages of
any character arising as a result of this License or out of the use or
inability to use the Work (including but not limited to damages for loss
of goodwill, work stoppage, computer failure or malfunction, or any and
all other commercial damages or losses), even if such Contributor has
been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty,
indemnity, or other liability obligations and/or rights consistent with
this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of
any other Contributor, and only if You agree to indemnify, defend, and
hold each Contributor harmless for any liability incurred by, or claims
asserted against, such Contributor by reason of your accepting any such
warranty or additional liability.

2. Unpack the archive into a directory.
3. Run the installation commands.
Note: Make sure you are a privileged user on the system.

perl Makefile.PL
make
test
make install

After installing, you can find documentation for this module with the perldoc command.

perldoc ServiceNow

Perl API architecture
This reference explains the Perl API architecture.
Figure 817: Perl API overview

ServiceNow.pm
createIncident()
queryTicket()
updateTask()
...

Incident.pm
insert()
query()
update()
...

Task.pm
insert()
query()
update()
...

GlideRecord.pm
insert()
query()
update()
...
**ServiceNow.pm**

At the top most of the class hierarchy is the **ServiceNow.pm** module. This module provides direct subroutines that delegate down into subroutines of the object heirarchy, providing a convenient calling convention if object oriented Perl programming is not a desired practice.

**ITIL Objects**

The next layer of the class hierarchy contains the ITIL Objects for example **Incident.pm**, and **Change.pm**. These objects extend the **Task.pm** object which inherits from the **GlideRecord.pm** object.

**Configuration.pm**

To use the Perl API, the programmer instantiates a **Configuration.pm** object and sets the service endpoint as well as the login credentials to use the API. For example:

```perl
my $CONFIG = ServiceNow::Configuration->new();
$CONFIG->setSoapEndpoint("https://instance_name.service-now.com/");
my $incident = ServiceNow::ITIL::Incident->new($CONFIG);
```

**Connection.pm**

The **Connection.pm** class defines the interfaces for making the SOAP calls to the platform. The default implementation uses the SOAP::Lite module, if you wish to implement your own SOAP interface, you can override this class with your own and implement the following subroutines:

- `new`
- `open`
- `send`
- `close`

**Note:**

If running behind a firewall, edit **Connection.pm** to specify information about the HTTP proxy.

In **Connection.pm**, change the line:

```perl
$me->{'SOAP'} = SOAP::Lite->proxy($CONFIG->getSoapEndPoint($target));
```

to:

```perl
$me->{'SOAP'} = SOAP::Lite->proxy($CONFIG->getSoapEndPoint($target),
    proxy => ['https' => 'http://myproxy.mycompany.com/']);
```

Substitute the appropriate proxy URL.

**Perl API classes**

The Perl API includes multiple classes.
# API

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceNow.pm</td>
<td>A class that contains data querying, creation, and modification subroutines for all the ITIL objects. It is a convenient entry point to the API without using the object oriented ITIL objects directly</td>
</tr>
</tbody>
</table>

## ITIL objects

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident.pm</td>
<td>The incident class, extends Task, which extends GlideRecord. Maps to the <strong>incident</strong> table.</td>
</tr>
<tr>
<td>Problem.pm</td>
<td>The problem class, extends Task, which extends GlideRecord. Maps to the <strong>problem</strong> table.</td>
</tr>
<tr>
<td>Change.pm</td>
<td>The change request class, extends Task, which extends GlideRecord. Maps to the <strong>change_request</strong> table.</td>
</tr>
<tr>
<td>Request.pm</td>
<td>The service request class, extends Task, which extends GlideRecord. Maps to the <strong>sc_request</strong> table.</td>
</tr>
<tr>
<td>RequestedItem.pm</td>
<td>The service request item class, extends Task, which extends GlideRecord. Maps to the <strong>sc_req_item</strong> table.</td>
</tr>
<tr>
<td>SC_Task.pm</td>
<td>The service request task class, extends Task, which extends GlideRecord. Maps to the <strong>sc_task</strong> table.</td>
</tr>
<tr>
<td>Task.pm</td>
<td>The task class extends GlideRecord. Maps to the <strong>task</strong> table.</td>
</tr>
<tr>
<td>Ticket.pm</td>
<td>The ticket class, extends Task, which extends GlideRecord. Maps to the <strong>ticket</strong> table.</td>
</tr>
</tbody>
</table>

## Other

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration.pm</td>
<td>The API configuration class, this class must be passed to the constructor of the other API classes. It defines the SOAP endpoint as well as the credentials to use for API access.</td>
</tr>
<tr>
<td>GlideRecord.pm</td>
<td>The class behind all table access classes.</td>
</tr>
<tr>
<td>Connection.pm</td>
<td>The Connection class implements the web service access layer in an interface that can be overridden. By default, it uses the SOAP::Lite package for web services support.</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Approval.pm</td>
<td>The approval class extends GlideRecord. Maps to the <code>sysapproval_approver</code> table.</td>
</tr>
<tr>
<td>Dictionary.pm</td>
<td>The dictionary class extends GlideRecord. Maps to the <code>sys_dictionary</code> table.</td>
</tr>
<tr>
<td>Attachment.pm</td>
<td>The attachment class extends GlideRecord. Used to add attachments to the other objects.</td>
</tr>
</tbody>
</table>

**Perl API ServiceNow**

The ServiceNow perl module is a collection of Perl subroutines that provides convenient and direct access to the ServiceNow platform.

**ServiceNow Perl API constructor**

This page lists the constructor for the ServiceNow Perl API.

### new

**new(Configuration object, optional Instance ID)**

Example:

```perl
$config = ServiceNow::Configuration->new();
$SN = ServiceNow->new($config);
```

Sets up an instance of the ServiceNow object using a Configuration object and an optional Instance ID.

**ServiceNow Perl API subroutines**

The ServiceNow Perl API includes multiple subroutines.

#### Incident

<table>
<thead>
<tr>
<th>createIncident</th>
<th>createIncident(optional parameters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>$number = $SN-&gt;createIncident({&quot;short_description&quot; =&gt; &quot;this is the short description&quot;);</td>
</tr>
<tr>
<td></td>
<td>Create an incident. Returns an incident number upon success. On failure returns undef.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>queryIncident</th>
<th>queryIncident(Reference to named parameters hash of Incident fields and exact values)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>@results = $SN-&gt;queryIncident({'number' =&gt; 'INC0000054'});</td>
</tr>
<tr>
<td></td>
<td>Query for Incidents matching specified criteria. Returns an array of incident records.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>updateIncident</td>
<td>Update a ServiceNow incident Returns undef on failure, all other values indicate success.</td>
</tr>
<tr>
<td>closeIncident</td>
<td>Close a ServiceNow Incident and optionally update field values. Returns the incident number on success, undef on failure.</td>
</tr>
<tr>
<td>reopenIncident</td>
<td>Reopen a closed ServiceNow incident. Returns the incident number on success, undef on failure.</td>
</tr>
<tr>
<td>reassignIncident</td>
<td>Reassign a ServiceNow Incident to a new assignment_group.(and optionally specify an assigned_to) Returns the incident number on success, undef on failure.</td>
</tr>
</tbody>
</table>

**updateIncident**

```perl
$ret = $SN->updateIncident($number, {$field => $value});
```

Example:

```
Update a ServiceNow incident
```

**closeIncident**

```perl
$number = $SN->closeIncident($number, {$field => $value});
```

Example:

```
Close a ServiceNow Incident and optionally update field values. Returns the incident number on success, undef on failure.
```

**reopenIncident**

```perl
my $ret = $SN->reopenIncident('INC99999');
```

Example:

```
Reopen a closed ServiceNow incident. Returns the incident number on success, undef on failure.
```

**reassignIncident**

```perl
my $ret = $SN->reassignIncident('INC99999', 'SOME_GROUP');
my $ret = $SN->reassignIncident('INC99999', 'SOME_GROUP', 'username');
```

Example:

```
Reassign a ServiceNow Incident to a new assignment_group.(and optionally specify an assigned_to) Returns the incident number on success, undef on failure.
```
## Ticket

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| createTicket | createTicket(Reference to named parameters hash of ticket fields and values) | Example: my $number = $SN-
>createTicket({"category" => "hardware"});Create a ServiceNow ticket associated with an incident. Returns a ticket number upon success. On failure returns undef. |
| updateTicket | updateTicket(The ticket number , Reference to named parameters hash of ticket fields and values to update) | Example: my $ret = $SN-
>updateTicket($number, {$field => $value});Returns undef on failure, all other values indicate success. |
| queryTicket  | queryTicket(Reference to named parameters hash of Incident fields and exact values) | Example: my @tickets = $SN-
>queryTicket({'number' => $number});Query for tickets matching specified criteria. Reference to array of hashes of all matching tickets, undef on failure or if no records found. |
| closeTicket  | closeTicket(Ticket number, {$field => $value})                                | Example: my $ret = $SN-
>closeTicket($number);
my $ret = $SN-
>closeTicket($number, {"comments" => "ticket closed"});Close a ServiceNow ticket and optionally specify work effort. Returns the ticket number on success, undef on failure. |
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reopenTicket</td>
<td>Reopen a closed ServiceNow ticket. Returns the ticket number on success, undef on failure.</td>
</tr>
<tr>
<td>Example:</td>
<td>my $ret = $SN- &gt;reopenTicket('TKT99999');</td>
</tr>
<tr>
<td>reassignTicket</td>
<td>Reassign a ServiceNow ticket to a new assignment_group. Returns the ticket number on success, undef on failure.</td>
</tr>
<tr>
<td>Example:</td>
<td>my $ret = $SN- &gt;reassignTicket('TKT99999', 'SOME_GROUP');</td>
</tr>
<tr>
<td></td>
<td>my $ret = $SN- &gt;reassignTicket('TKT99999', 'SOME_GROUP', 'username');</td>
</tr>
</tbody>
</table>

### RequestedItem / Request

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createRequestedItem</td>
<td>Create a Service Catalog RequestedItem (and indirectly the associated Request and Tasks). Returns a RequestedItem number upon success. On failure returns undef.</td>
</tr>
<tr>
<td>Example:</td>
<td>my $req = $SN- &gt;createRequestedItem('CITM10000', 'username', {'description' =&gt; 'Some description', 'group' =&gt; 'SOME_GROUP'});</td>
</tr>
</tbody>
</table>
### queryRequestedItem

**queryRequestedItem** *(Reference to named parameters hash of RequestedItem fields and exact values)*

**Example:**

```perl
my $requestedItems = $SN->queryRequestedItem({'number' => 'SOME_RI_NUMBER'});
```

Query for RequestedItems matching specified criteria. Reference to array of hashes of all matching RequestedItem, undef on failure or if no records found.

### queryRequest

**queryRequest** *(Reference to named parameters hash of Request fields and exact values)*

**Example:**

```perl
my $requests = $SN->queryRequest({'number' => 'SOME_REQUEST_NUMBER'});
```

Query for Requests matching specified criteria. Reference to array of hashes of all matching Request, undef on failure or if no records found.

---

## Task

### createTask

**createTask** *(optional parameters)*

**Example:**

```perl
$number = $SN->createTask({"short_description" => "this is the short description"});
```

Create a task record. Returns a task number upon success. On failure returns undef.

### closeTask

**closeTask** *(task number, optional work effort in seconds)*

**Example:**

```perl
my $ret = $SN->closeTask($number);
my $ret = $SN->closeTask($number, {field => $value});
```

Close a ServiceNow Task and optionally update field values. Returns true on success, undef on failure.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reopenTask</td>
<td>Reopen a closed ServiceNow Task. Returns true on success, undef on failure.</td>
</tr>
<tr>
<td>reassignTask</td>
<td>Reassign a ServiceNow Task to a new assignment_group. Returns true on success, undef on failure.</td>
</tr>
<tr>
<td>updateTask</td>
<td>Update a ServiceNow Task. Returns true on success, undef on failure.</td>
</tr>
</tbody>
</table>

**Example for reopenTask**

```perl
my $ret = $SN->reopenTask('TASK99999');
```

**Example for reassignTask**

```perl
my $ret = $SN->reassignTask('TASK99999', 'SOME_GROUP');
my $ret = $SN->reassignTask('TASK99999', 'SOME_GROUP', 'username');
```

**Example for updateTask**

```perl
my $ret = $SN->updateTask($number, {$field => $value});
```
### queryTask

queryTask(reference to named parameters hash of Task fields and exact values)

Example:

```perl
my @tasks = $SN->queryTask({'number' => $number});
foreach my $task (@tasks) {
    print "Incident number: $task->{'number'}\n";
    print "Assignent Group: $task->{'assignment_group'}\n";
    print "Opened by: $task->{'opened_by'}\n";
    print "SD: $task->{'short_description'}\n";
    print "TW: $task->{'time_worked'}\n";
}
```

Query for Tasks matching specified criteria. Array of hashes of all matching Tasks, undef on failure or if no records found.

### Journal

#### queryJournal

queryJournal(Incident/Ticket/Task number, optional journal field name)

Query for journals entries for the specified incident, ticket or task. Return array of hashes of all matching Journals, undef on failure or if no records found. There will be one hash per per journal entry, 'value' will contain the journal entry string, 'element' will be the name of the field (e.g. 'comments', 'work_notes', etc.)

#### appendJournal

appendJournal(Incident/Ticket/Task number, field name, journal text)

Example:

```perl
my $ret = $SN->appendJournal("INC99999", 'comments', "some comment text");
```

Append a journal entry to the specified journal field of an incident, ticket, or task. Returns true on success, undef on failure.
### Approval

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryApproval</td>
<td>queryApproval(reference to named parameters hash of approval fields and exact values)</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>my @approvals = $SN-&gt;queryApproval({'approver' =&gt; 'username'});</td>
</tr>
<tr>
<td></td>
<td>Query for approvals Return array of hashes of all matching approvals, undef on failure or if no records found.</td>
</tr>
<tr>
<td>approve</td>
<td>approve(sys_id of the approval, approval state, optional comment text)</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>my $ret = $SN-&gt;approve($sys_id, 'Approved');</td>
</tr>
<tr>
<td></td>
<td>my $ret = $SN-&gt;approve($sys_id, 'Rejected', &quot;Please do something else&quot;);</td>
</tr>
</tbody>
</table>

### Dictionary

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryFields</td>
<td>queryFields(table, optional boolean)</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>my @fields = $SN-&gt;queryFields('incident');</td>
</tr>
<tr>
<td></td>
<td>List all the fields of an Incident, Request, RequestedItem or Task. Returns a reference to a hash of fields in the specified table type. The hash key is the field name, and the hash value is a hash reference to attributes about the field: 'mandatory', 'hint', 'label', 'reference' and 'choice'. Returns undef on failure. If getchoices is true then 'choices' is a reference to a hash containing individual choices, keyed by choice value and containing choice 'label' and 'hint'.</td>
</tr>
</tbody>
</table>
Creating an Incident

```perl
#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
my $CONFIG = ServiceNow::Configuration->new();
$CONFIG->setSoapEndPoint("https://demoi1.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");
my $SN = ServiceNow->new($CONFIG);
my $number = $SN->createIncident(
    "short_description" => "this incident was created from the Perl API", "category" => "hardware") ;
print $number ."\n";
```

Querying an Incident

```perl
#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
my $CONFIG = ServiceNow::Configuration->new();
$CONFIG->setSoapEndPoint("https://demoi1.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");
my $SN = ServiceNow->new($CONFIG);
my @incidents = $SN->queryIncident({'number' => 'INC00002'});
my $count = scalar(@incidents);
print "number of incidents=": $count ."\n";
foreach my $incident (@incidents) {
    print "Incident number: $incident->{'number'}\n";
    print "Assignent Group: $incident->{'assignment_group'}\n";
    print "Opened by: $incident->{'opened_by'}\n";
    print "Opened by DV: $incident->{'dv_opened_by'}\n";
    print "SD: $incident->{'short_description'}\n";
    print "TW: $incident->{'time_worked'}\n";
    print "\n";
}
```

Querying Journal fields

```perl
#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
my $CONFIG = ServiceNow::Configuration->new();
```
Perl API GlideRecord

ServiceNow Perl API - GlideRecord perl module is an object representation of a GlideRecord object used to access your ServiceNow instance.

### Constructor

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>new(Configuration object, Table name, optional caller object)</td>
</tr>
</tbody>
</table>

```perl
$config = ServiceNow::Configuration->new();
$glideRecord = ServiceNow::GlideRecord->new($config, 'incident', $me);
```

Constructor. Access to the ServiceNow Glide Record object. The caller object is optional unless creating a Class that inherits GlideRecord (See any class in ServiceNow/ITIL for example).

### Subroutines

<table>
<thead>
<tr>
<th>Subroutines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>insert</td>
<td>insert(optional hash argument)</td>
</tr>
</tbody>
</table>

Example:

```perl
$glideRecord->insert();
```

Inserts glide record into Table. Returns sys id.

<table>
<thead>
<tr>
<th>setValue</th>
<th>setValue(name, value)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>$glideRecord-&gt;setValue('caller_id', '56');</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subroutines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addQuery</td>
<td>Sets element within Glide Record with name to specified value. Will not effect the GlideRecord within the Table until inserted or updated.</td>
</tr>
<tr>
<td>addQuery</td>
<td>addQuery(name, value)</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>$glideRecord-&gt;addQuery('number','INC1000014');</td>
</tr>
<tr>
<td></td>
<td>$glideRecord-&gt;query();</td>
</tr>
<tr>
<td></td>
<td>Refines query to include only the Glide Records with field name=value.</td>
</tr>
<tr>
<td>query</td>
<td>query(optional hash arguments)</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>$glideRecord-&gt;query();</td>
</tr>
<tr>
<td></td>
<td>Returns all Glide Records in the Table with specified query. Step through the Records with the next() call.</td>
</tr>
<tr>
<td>next</td>
<td>next()</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>if($glideRecord-&gt;next());</td>
</tr>
<tr>
<td></td>
<td>while($glideRecord-&gt;next());</td>
</tr>
<tr>
<td></td>
<td>Steps through the results of Glide Record query. Returns TRUE if more elements exist.</td>
</tr>
<tr>
<td>update</td>
<td>update(optional hash arguments)</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>$glideRecord-&gt;setValue('name','value');</td>
</tr>
<tr>
<td></td>
<td>$glideRecord-&gt;update();</td>
</tr>
<tr>
<td></td>
<td>Updates Glide Record in table with the Glide Record object. Changes to Glide Record object will not take effect until updated or inserted. Returns sys_id of record on success, undef of failure.</td>
</tr>
<tr>
<td>getValue</td>
<td>getValue(name)</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>$glideRecord-&gt;getValue($name);</td>
</tr>
<tr>
<td></td>
<td>Get value of element name in GlideRecord. Returns string value of element.</td>
</tr>
</tbody>
</table>
Subroutines

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>getDisplayValue</td>
<td>getDisplayValue(name)</td>
<td>$glideRecord-&gt;getDisplayValue($name);</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Gets display value of element name in GlideRecord. A display value would be the string name, instead of the sys_id in the case of a reference field, or the string value instead of the number value in the case of choice fields.</td>
</tr>
</tbody>
</table>

Perl API task

ServiceNow Perl API - task module is an object representation of a Task in the ServiceNow platform.

The Perl API task provides subroutines for querying, updating, and creating tasks. Task is the parent class of Incident, Problem, Change, SC_Task and Ticket. These child classes inherit subroutines from this class.

Constructor

new(Configuration);

Example:

```perl
$task = ServiceNow::ITIL::Task->new($CONFIG);
```

This example takes a configuration object and manufactures a Task object connected to the ServiceNow instance.

Perl API task subroutines

The Perl API task provides subroutines for querying, updating, and creating tasks.

Subroutines

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>create(optional parameters)</td>
<td>$task-&gt;create();</td>
</tr>
<tr>
<td></td>
<td>Creates and inserts Task/Incident/Ticket/SC_Task record into the respective table. Returns the number of created record on success, undef on failure.</td>
<td></td>
</tr>
<tr>
<td>close</td>
<td>close(number of record, optional parameters)</td>
<td>$task-&gt;close('INC1000312');</td>
</tr>
<tr>
<td></td>
<td>Sets the state of Task/Incident/Ticket/SC_Task to closed and updates the respective table. Returns the number of created</td>
<td></td>
</tr>
</tbody>
</table>
### Perl API Incident

An object representation of an incident in the ServiceNow platform. Provides subroutines for querying, updating, and creating incidents.

#### Incident module

ServiceNow Perl API - Incident perl module

#### Creating an Incident

```perl
#!/usr/bin/perl -w

use ServiceNow;
use ServiceNow::Configuration;
use ServiceNow::ITIL::Incident;
```
my $CONFIG = ServiceNow::Configuration->new();

$CONFIG->setSoapEndPoint("https://demoi1.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");

# setting incident values as a hash map in the insert argument
my $incident = ServiceNow::ITIL::Incident->new($CONFIG);
my $sys_id = $incident->insert({"short_description" => "this incident was created from the Perl API", "category" => "hardware"});
print $sys_id . "\n";

# setting incident values by making setValue calls to the incident object
$incident = ServiceNow::ITIL::Incident->new($CONFIG);
$incident->setValue("short_description", "this incident was created from the Perl API - 2");
$incident->setValue("category", "hardware");
$sys_id = $incident->insert();
print $sys_id . "\n";

### Querying for Incidents

#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
use ServiceNow::ITIL::Incident;

my $CONFIG = ServiceNow::Configuration->new();

$CONFIG->setSoapEndPoint("https://demoi1.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");

my $incident = ServiceNow::ITIL::Incident->new($CONFIG);
$incident->addQuery("assignment_group", "Service Desk");
$incident->addQuery("category", "Hardware");
$incident->query();
while($incident->next()) {
    print "number=" . $incident->getValue("number") . "\n";
    print "sd=" . $incident->getValue("short_description") . "\n";
    print "opened_by Display Value= " . $incident->getDisplayValue("opened_by") . "\n";
    print "opened_by sys_id= " . $incident->getValue('opened_by');
}

### Adding an attachment to a newly created Incident

#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
use ServiceNow::ITIL::Incident;

my $CONFIG = ServiceNow::Configuration->new();

$CONFIG->setSoapEndPoint("https://demoi1.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");

my $incident = ServiceNow::ITIL::Incident->new($CONFIG);
$incident->setValue("short_description", "test incident for attachment 2");
$incident->insert();
$incident->attach("/Users/davidloo/Desktop/test_files/number_test.xls");

**Constructor**

**new**

ew(Configuration);

Example:

$incident = ServiceNow::ITIL::Incident->new($CONFIG);

Takes a configuration object and manufactures an Incident object connected to the ServiceNow instance.

**Subroutines inherited from Task.pm**

- attach
- close
- create
- queryJournal
- reassign
- reopen

**Subroutines inherited from GlideRecord.pm**

- addQuery
- getValue
- getDisplayValue
- setValue
- next
- insert
- query
- update

**Subroutines**

**close**

close(number, hashmap);
Example:

```perl
$incident->close($number)
```

Close an incident and update values described in the hash map passed in.

### reopen

`reopen(number, hashmap);`

Example:

```perl
$incident->reopen($number);
```

Re-open a closed incident and update values described in the hash map passed in.

### createProblem

`createProblem();`

Create a problem ticket from an incident and associate it. Returns the sys_id of the newly created problem ticket.

### createChange

`createChange();`

Create a change request from an incident and associate it. Returns the sys_id of the newly created change request.

**Perl API change**

An object representation of a change in the ServiceNow platform. Provides subroutines for querying, updating, and creating change requests.

### Constructor

`new`

`new(Configuration);`

Example:

```perl
$change = ServiceNow::ITIL::Change->new($CONFIG);
```

Takes a configuration object and manufactures an Change object connected to the ServiceNow instance.

### Subroutines inherited from Task.pm

- attach
- close
- create
- queryJournal
- reassign
• reopen

Subroutines inherited from GlideRecord.pm

• addQuery
• getValue
• getDisplayValue
• setValue
• next
• insert
• query
• update

PerI API problem
An object representation of a problem in the ServiceNow platform. Provides subroutines for querying, updating, and creating problems.

Constructor - new

new(Configuration);

Example:

$problem = ServiceNow::ITIL::Problem->new($CONFIG);

Takes a configuration object and manufactures a Problem object connected to the ServiceNow instance.

Subroutines inherited from Task.pm

• attach
• close
• create
• queryJournal
• reassign
• reopen

Subroutines inherited from GlideRecord.pm

• addQuery
• getValue
• getDisplayValue
• setValue
• next
• insert
• query
• update

Perl API request
An object representation of a request in the ServiceNow platform. Provides subroutines for querying, updating, and creating service catalog requests.

**Description**

An object representation of a request in the ServiceNow platform. Provides subroutines for querying, updating, and creating service catalog requests.

**Constructor**

```
new
new(Configuration);
```

Example:

```
$request = ServiceNow::ITIL::Request->new($CONFIG);
```

Takes a configuration object and manufactures a Request object connected to the ServiceNow instance.

**Subroutines inherited from Task.pm**

- attach
- close
- create
- queryJournal
- reassign
- reopen

**Subroutines inherited from GlideRecord.pm**

- addQuery
- getValue
- getDisplayValue
- setValue
- next
- insert
- query
- update

**Subroutines**

**createRequest**

```
createRequest(user);
```

Create a service request for the specified user.
Perl API requested item

<table>
<thead>
<tr>
<th>ServiceNow Perl API Module</th>
<th>Description</th>
<th>Constructor</th>
<th>Subroutines inherited from Task.pm</th>
<th>Subroutines inherited from GlideRecord.pm</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestedItem</td>
<td>An object representation of a Requested Item in the ServiceNow platform. Provides subroutines for querying, updating, and creating Service Catalog requested item.</td>
<td>new(Configuration); Example: $req_item = ServiceNow::ITIL::RequestedItem-&gt;new($CONFIG);</td>
<td>attach, close, create, queryJournal, reassign, reopen</td>
<td>attach, close, create, query, update</td>
</tr>
</tbody>
</table>

**System Requirements**
The ServiceNow Perl API requires Perl 5.8 with the following modules installed:
- SOAP::Lite (prerequisites [http://soaplite.com/prereqs.html](http://soaplite.com/prereqs.html)) 0.71 or later
- Crypt::SSLeay
- IO::Socket::SSL

**Perl API dictionary**
An object representation of a dictionary record in the ServiceNow platform. Provides subroutines for querying, updating, and creating sys_dictionary.

**Dictionary module**
ServiceNow Perl API - dictionary module

**new**

new(Configuration);
Example:

```perl
$dic = ServiceNow::ITIL::Dictionary->new($CONFIG);
```

Takes a configuration object and manufactures a Dictionary object connected to the ServiceNow instance
queryFields

queryFields(table name, optional boolean choice)

Example:

```perl
$sys_dictionary->queryFields('incident');
```

Returns an array of hashes containing each field for specified table. If choice is specified then get choice values for each of the choice fields and put them into the hash element 'choices'.

Perl API configuration

<table>
<thead>
<tr>
<th>ServiceNow Perl API Module</th>
<th>Description</th>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket Perl</td>
<td>An object representation of a Configuration object used to access your ServiceNow instance.</td>
<td><code>new();</code></td>
</tr>
<tr>
<td>System Requirements</td>
<td>The ServiceNow Perl API requires Perl 5.8 with the following modules installed:</td>
<td><code>new();</code></td>
</tr>
<tr>
<td></td>
<td>• SOAP::Lite (prerequisites <a href="http://soaplite.com/prereqs.html">http://soaplite.com/prereqs.html</a> 0.71 or later</td>
<td><code>new();</code></td>
</tr>
<tr>
<td></td>
<td>• Crypt::SSLeay</td>
<td><code>new();</code></td>
</tr>
<tr>
<td></td>
<td>• IO::Socket::SSL</td>
<td><code>new();</code></td>
</tr>
</tbody>
</table>

Create a new Configuration object and start customizing it to be used for other objects.
Table 1084: Subroutines

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getSoapEndPoint(target_table)</td>
<td>Gets the complete SOAP endpoint used to access your ServiceNow instance, given the table name. For example:</td>
</tr>
<tr>
<td>setSoapEndPoint(endpoint_url)</td>
<td>Sets the complete SOAP endpoint used to access your ServiceNow instance. For example:</td>
</tr>
<tr>
<td>getUserName()</td>
<td>Get the user name used to authenticate a connection to the SOAP endpoint.</td>
</tr>
<tr>
<td>setUserName(user_name)</td>
<td>Set the user name used to authenticate a connection to the SOAP endpoint.</td>
</tr>
<tr>
<td>getUserPassword()</td>
<td>Get the user password used to authenticate a connection to the SOAP endpoint.</td>
</tr>
<tr>
<td>setUserPassword(user_password)</td>
<td>Set the user password used to authenticate a connection to the SOAP endpoint.</td>
</tr>
<tr>
<td>getConnection(target_table)</td>
<td>Get the Connection object used to access the ServiceNow SOAP endpoint.</td>
</tr>
</tbody>
</table>

Perl API attachment
The ServiceNow Perl API - Attachment perl module is an object representation of an Attachment in the ServiceNow platform. Provides subroutines for creating an attachment and attaching to an existing record.

Constructor - new

ew(Configuration);

Example:

$task = ServiceNow::Attachment->new($CONFIG);

Takes a configuration object and manufactures an Task object connected to the ServiceNow instance

Subroutines - create

create(path, table_name, sys_id)  

Example:

$attachment->create("/Users/davidloo/Desktop/test_files/number_test.xls", "incident", "9d385017c611228701d22104cc95c371");
Creates an attachment from a file on the local disk, to an existing record defined by table_name and sys_id
Returns the sys_id of the ecc_queue record, undef if failed

Perl API service catalog task

<table>
<thead>
<tr>
<th>ServiceNow Perl API Module</th>
<th>Description</th>
<th>Constructor</th>
<th>Subroutines inherited from Task.pm</th>
<th>Subroutines inherited from GlideRecord.pm</th>
</tr>
</thead>
</table>
| SC_Task                    | An object representation of a Service Request Task in the ServiceNow platform. Provides subroutines for querying, updating, and creating sc_task. | new(Configuration); Example:

```perl
$sc_task = ServiceNow::ITIL::RequestedItem->new($CONFIG);
```
| attach                      | addQuery    |
| close                       | getValue    |
| create                      | getDisplayValue |
| queryJournal                | setValue    |
| reopen                      | next        |
|                             | insert      |
|                             | query       |
|                             | update      |

System Requirements

The ServiceNow Perl API requires Perl 5.8 with the following modules installed:

- SOAP::Lite (prerequisites http://soaplite.com/prereqs.html) 0.71 or later
- Crypt::SSLLeay
- IO::Socket::SSL

Perl API ticket

<table>
<thead>
<tr>
<th>ServiceNow Perl API Module</th>
<th>Description</th>
<th>Constructor</th>
<th>Subroutines inherited from Task.pm</th>
<th>Subroutines inherited from GlideRecord.pm</th>
</tr>
</thead>
</table>
| Ticket                     | An object representation of a Ticket in the ServiceNow platform. Provides subroutines for querying, updating, and creating tickets. | new(Configuration); Example:

```perl
$ticket = ServiceNow::ITIL::Ticket->new($CONFIG);
```
| attach                      | addQuery    |
| close                       | getValue    |
| create                      | getDisplayValue |
| queryJournal                | setValue    |
| reopen                      | next        |
|                             | insert      |
|                             | query       |
|                             | update      |
## Approval

An object representation of an Approval record in the ServiceNow instance. Provides subroutines for querying, updating, and creating approvals.

### System Requirements

The ServiceNow Perl API requires Perl 5.8 with the following modules installed:

- SOAP::Lite (prerequisites [http://soaplite.com/prereqs.html](http://soaplite.com/prereqs.html)) 0.71 or later
- Crypt::SSL
- IO::Socket::SSL

### Constructor

```perl
new(Configuration);
```

**Example:**

```perl
$approvals = ServiceNow::ITIL::Approval->new($CONFIG);
```

Takes a configuration object and manufactures an Approval object connected to the ServiceNow instance.
Table 1085: Subroutines

<table>
<thead>
<tr>
<th>Subroutines inherited from Task.pm</th>
<th>Subroutines inherited from GlideRecord.pm</th>
<th>reject</th>
<th>approve</th>
</tr>
</thead>
<tbody>
<tr>
<td>• attach</td>
<td>• addQuery</td>
<td>reject(sys_id of approval record, optional comment string)</td>
<td>approve(sys_id of approval record, optional comment string)</td>
</tr>
<tr>
<td>• close</td>
<td>• getValue</td>
<td>Example: $approval-&gt;reject($sysID, $comment);</td>
<td>Example: $approval-&gt;approve($sysID, $comment);</td>
</tr>
<tr>
<td>• create</td>
<td>• getDisplayValue</td>
<td>Reject the Approval record with specified sys_id, and add comment to approval if given.</td>
<td>Approve the Approval record with specified sys_id, and add comment to approval if given.</td>
</tr>
<tr>
<td>• queryJournal</td>
<td>• setValue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• reassign</td>
<td>• next</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• reopen</td>
<td>• insert</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• query</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• update</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Perl API examples
This reference provides examples of the Perl API.

Using the ServiceNow.pm module

Creating an incident:

```perl
#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
my $CONFIG = ServiceNow::Configuration->new();
$CONFIG->setSoapEndPoint("https://<instance name>.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");
my $SN = ServiceNow->new($CONFIG);
my $number = $SN->createIncident({"short_description" => "this incident was created from the Perl API", "category" => "hardware"});
print $number . "\n";
```

Querying an Incident:

```perl
#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
my $CONFIG = ServiceNow::Configuration->new();
$CONFIG->setSoapEndPoint("https://<instance name>.service-now.com/");
$CONFIG->setUserName("admin");
```
$CONFIG->setUserPassword("admin");

my $SN = ServiceNow->new($CONFIG);

my @incidents = $SN->queryIncident({'number' => 'INC00002'});
my $count = scalar(@incidents);
print "number of incidents=
" . $count . "\n"

foreach my $incident (@incidents) {
    print "Incident number: $incident->{'number'}\n";
    print "Assignent Group: $incident->{'assignment_group'}\n";
    print "Opened by: $incident->{'opened_by'}\n";
    print "Opened by DV: $incident->{'dv_opened_by'}\n";
    print "SD: $incident->{'short_description'}\n";
    print "TW: $incident->{'time_worked'}\n";
}

Querying Journal fields:

#!/usr/bin/perl -w

use ServiceNow;
use ServiceNow::Configuration;

my $CONFIG = ServiceNow::Configuration->new();

$CONFIG->setSoapEndPoint("https://<instance name>.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");

my $SN = ServiceNow->new($CONFIG);

my @journals = $SN->queryJournal('INC00002');

print $journals[0]->{'element'} . " = " . $journals[0]->{'value'} . "\n"
print $journals[1]->{'element'} . " = " . $journals[1]->{'value'} . "\n";

Using the ITIL Objects

Creating an Incident:

#!/usr/bin/perl -w

use ServiceNow;
use ServiceNow::Configuration;
use ServiceNow::ITIL::Incident;

my $CONFIG = ServiceNow::Configuration->new();

$CONFIG->setSoapEndPoint("https://<instance name>.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");

# setting incident values as a hash map in the insert argument
my $incident = ServiceNow::ITIL::Incident->new($CONFIG);
my $sys_id = $incident->insert({"short_description" => "this incident was created from the Perl API", "category" => "hardware"});
print $sys_id . "\n";

# setting incident values by making setValue calls to the incident object
$incident = ServiceNow::ITIL::Incident->new($CONFIG);
$incident->setValue("short_description", "this incident was created from the Perl API - 2");
$incident->setValue("category", "hardware");
$sys_id = $incident->insert();
print $sys_id . "\n";

Querying for Incidents:

#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
use ServiceNow::ITIL::Incident;

my $CONFIG = ServiceNow::Configuration->new();
$CONFIG->setSoapEndPoint("https://<instance name>.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");

my $incident = ServiceNow::ITIL::Incident->new($CONFIG);
$incident->addQuery("assignment_group", "Service Desk");
$incident->addQuery("category", "Hardware");
$incident->query();
while($incident->next()) {
    print "number=" . $incident->getValue("number") . "\n";
    print "sd=" . $incident->getValue("short_description") . "\n";
    print "opened_by Display Value= " . $incident->getDisplayValue("opened_by") . "\n";
    print "opened_by sys_id= " . $incident->getValue('opened_by');
}

Adding an attachment to a newly created Incident:

#!/usr/bin/perl -w
use ServiceNow;
use ServiceNow::Configuration;
use ServiceNow::ITIL::Incident;

my $CONFIG = ServiceNow::Configuration->new();
$CONFIG->setSoapEndPoint("https://<instance name>.service-now.com/");
$CONFIG->setUserName("admin");
$CONFIG->setUserPassword("admin");

my $incident = ServiceNow::ITIL::Incident->new($CONFIG);
$incident->setValue("short_description", "test incident for attachment 2");
$incident->insert();
$incident->attach("/Users/davidloo/Desktop/test_files/number_test.xls");

Scripted SOAP web services

Scripted SOAP Web Services allow a ServiceNow administrator to create new web services that are not addressed by the system.
You can define input and output parameters for the web service and use JavaScript to perform operations. Though this feature is very powerful, use Direct Web Services or Web Service import sets instead whenever possible since they are simpler to implement and maintain.
Security

When *strict security* is enforced on a system, the HTTP authenticated user must have the *soap_script* role to execute the scripted web service.

Creating a new web service

When the Web Services Provider - Scripted plugin is activated, a new module Scripted Web Services is available under the System Web Services application.

![System Web Services](image)

**Figure 818: Scripted web service module**

Click the module to display a list of example scripted Web Services.

<table>
<thead>
<tr>
<th>Name</th>
<th>Active</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetProperty</td>
<td>true</td>
<td>Get a property value</td>
</tr>
<tr>
<td>GetTransactionCount</td>
<td>true</td>
<td>Get the number of transactions</td>
</tr>
<tr>
<td>OrderBlackBerry</td>
<td>true</td>
<td>Order a BlackBerry</td>
</tr>
</tbody>
</table>

**Figure 819: Sys web service**

**Example 1: Retrieving a system property**

The first step is to define the incoming and return parameters. This is done by adding an entry to the Input Parameters and Output Parameters. These parameters are used to construct and present a meaningful WSDL, and they do not add to the functionality of processing the actual Web Service itself.
The parameters are referenced in the script of the Web Service. Any of the input parameters are retrieved using the following syntax:

```javascript
var a = request.property;
```

The output parameters are set by using the following syntax:

```javascript
response.property = "ABC";
```

The following example demonstrates how to retrieve a system property and return it as part of the SOAP response. The example shows how to create a custom scripted Web Service to do something specific that the base ServiceNow system direct Web Services cannot.
Figure 821: Get property

Example 2: Ordering a Blackberry

Direct Web Services in ServiceNow operate on tables and their data, while the following example shows how to initiate a business solution, such as ordering a Blackberry, by invoking a scripted Web Service. The following input and output parameters will support the Blackberry example:
This script shows how to use the above parameters to add a Blackberry to the service catalog shopping cart and order it. The request number is returned in the request_number field of the SOAP response.

```javascript
var cart = new Cart();
var item = cart.addItem('e2132865c0a8016500108d9cee411699');
cart.setVariable(item, 'original', request.phone_number);

// set the requested for
var gr = new GlideRecord("sys_user");
gr.addQuery("user_name", request.requested_for);
gr.query();
if (gr.next()) {
    var cartGR = cart.getCart();
    cartGR.requested_for = gr.sys_id;
    cartGR.update();
}

var rc = cart.placeOrder();
response.request_number = rc.number;
```

**Global Variables**

To facilitate custom processing of incoming SOAP requests, the following global variables are available in the script context:

2. `soapRequestXML`: a string object representing the incoming SOAP envelope XML.
3. `request`: a Javascript object containing mapped values (mapped to input parameter names) of the incoming SOAP envelope.
4. `response`: a Javascript object which allows the script programmer to customize the response values. See [Customize Response](#)
Customize response
Follow this example in order to customize and have control over the XML payload of the SOAP response.

1. Create a customized XML document using the `XMLDocument` script include object.
2. Set its document element to the variable `response.soapResponseElement` in a scripted web service.

For example, the following scripted web service script:

```javascript
var xmldoc = new XMLDocument("<myResponse></myResponse>");
xmldoc.createElement("element_one", "test"); // creates the new element at the document element level if setCurrent is never called
xmldoc.createElement("element_two", "new2 value"); // calling without a value will create a new element by itself
var el = xmldoc.createElement("element_three");
xmldoc.setCurrent(el); // this is now the parent of any new elements created subsequently using createElement()
xmldoc.createElement("newChild", "test child element");
response.soapResponseElement = xmldoc.getDocumentElement();
```

Is used to accept the following request:

```xml
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:tes="http://www.service-now.com/TestCustomResponse">
 <soapenv:Header/>
 <soapenv:Body>
  <tes:execute/>
 </soapenv:Body>
</soapenv:Envelope>
```

Which will respond with the following SOAP response:

```xml
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:tes="http://www.service-now.com/TestCustomResponse">
 <soapenv:Header/>
 <soapenv:Body>
  <myResponse>
   <element_one>test</element_one>
   <element_two>new2 value</element_two>
   <element_three>
    <newChild>test child element</newChild>
   </element_three>
  </myResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

WSDL support will need to be created externally. The SOAP endpoint will need to be referred back to the scripted web service in question.

Static WSDL
A static WSDL allows you to define a custom WSDL for web service requests.

Certain web service clients require SOAP access to ServiceNow through a specific Web Service Description Language (WSDL) format. This required format may differ from the standard ServiceNow WSDL format. In these cases you can create a static WSDL that matches the required format.

Create a scripted web service
To use a static WSDL, create a scripted web service.
Role required: web_service_admin or admin

1. Navigate to **System Web Service > Scripted Web Services.**
2. Click **New.**
3. Enter a **Name** for the scripted web service such as FakeStockValue.
4. Enter a **Script** for the web service to run.
5. Click **Submit.**

Scripted web service example
This example demonstrates the processing script for the FakeStockValue web service.

```javascript
var vProcessor = new FakeStockValue(soapRequestXML);

var responseElement = vProcessor.process(); if (responseElement != null) {
    response.soapResponseElement = responseElement;
} else {
    response.soapResponseElement = vProcessor.generateSoapFault("unknown error");
}
```

Create a static WSDL
Create a static WSDL with the required format to override the standard WSDL for your scripted web service.

Role required: web_service_admin or admin

1. Navigate to **System Web Services > Static WSDL.**
2. Create a static WSDL record using the same name as the scripted web service, such as FakeStockValue.
3. Enter the custom WSDL into the **WSDL** field.
4. Click **Submit.**

Static WSDL example
This example demonstrates the FakeStockValue WSDL.

```xml
<?xml version= "1.0" ?>
<definitions name = "StockQuote"
targetNamespace = "http://example.com/stockquote.wsdl"
xmlns:tns = "http://example.com/stockquote.wsdl"
xmlns:xsd1 = "http://example.com/stockquote.xsd"
xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
xmlns = "http://schemas.xmlsoap.org/wsdl/"
>

<types><schema targetNamespace = "http://example.com/stockquote.xsd"
xmlns = "http://www.w3.org/2000/10/XMLSchema">
<element name = "TradePriceRequest">
<complexType><all>
<element name = "tickerSymbol" type = "string" /></all></complexType>
</element>
<element name = "TradePrice">
<complexType><all>
<element name = "price" type = "float" /></all></complexType>
</element>
</schema></types>

<message name = "GetLastTradePriceInput">
<part name = "body" element = "xsd1:TradePriceRequest" />
</message>

<message name = "GetLastTradePriceOutput">
<part name = "body" element = "xsd1:TradePrice" />
</message>

<portType name = "StockQuotePortType">
<operation name = "GetLastTradePrice">
<input message = "tns:GetLastTradePriceInput" />
<output message = "tns:GetLastTradePriceOutput" />
</operation>
</portType>

<binding name = "StockQuoteSoapBinding" type = "tns:StockQuotePortType">
<soap:binding style = "document" transport = "http://schemas.xmlsoap.org/"

```
Create a static WSDL script include

Create a script include to define the majority of the code used to process static WSDL requests.

Role required: script_include_admin or admin

By implementing the majority of the custom functionality in a script include, you can reuse the script include in multiple areas.

1. Navigate to System UI > Script Includes.
2. Click New.
3. Enter a Name for the script include that matches the name of the static WSDL, such as FakeStockValue.
4. Enter the script include code in the Script field.
5. Click Submit.

Static WSDL script include example

This example demonstrates the FakeStockValue script include that implements much of the static WSDL behavior.

```javascript
var FakeStockValue = Class.create();

FakeStockValue.prototype = {
  initialize : function(requestXML) {
    //Use some backend XML utilities...you could use string tools if you wish
    this.xmlutil = Packages.com.glide.util.XMLUtil;
    //converting the string to an XML Document
    this.fSoapDoc = new XMLDocument(requestXML);
  },

  process : function() {
    var soapBody = this.fSoapDoc.getNode("/Envelope/Body");
    //Our WSDL was formatted to have the only first child element be the function
    var funcNode = this.xmlutil.getFirstChildElement(soapBody);
    var nodeName = this.xmlutil.getNodeNameNS(funcNode);

    //If the function for this SOAP request is TradePriceRequest, perform the necessary actions
    if (nodeName == "TradePriceRequest") {
      return this.fakeOutTradePriceRequest(funcNode);
    }

    //Couldn't find any supported functions in this SOAP request
    return this.generateSoapFault("un-supported API call: " + nodeName);
  },

  fakeOutTradePriceRequest : function (funcNode) {
    //Create the beginnings of our XML response
  }

};
```
var r = new XMLDocument("<GetLastTradePriceOutput xmlns='https://www.service-now.com/vws/FakeStockValue'/>");

// Do the necessary actions here...we're going to get the USER ID of the user
// Used to make this SOAP call. Then we will return the stock symbol they were asking about
var usersysid = gs.getUserID();
var gr = new GlideRecord("sys_user");
gr.get(usersysid);
var username = gr.user_name;
var quoteSymbol = this.xmlutil.getText(funcNode);
// Create a "message" element to store our response message
var message = username + ", You were looking for a quote on "+quoteSymbol;
return r.createElement("message", message);

}

initialize function

The initialize function takes the XML request string and converts it to an XML Document object that you can navigate and manipulate using libraries. Alternatively, you can leave the XML request as a string and navigate it using regular expressions.

process function

The process function is called by the scripted web service. This function grabs the first child element in the XML after the body element. The WSDL uses this child element to determine which function to use. In this WSDL there is only one possible function but most WSDLs provide many functions. If more functions were available, there would be more "if" statements that tested the first child element for the various function names.

fakeOutTradePriceRequest function

The fakeOutTradePriceRequest function is the implementation of the only available function in the WSDL. This function looks up the user that the SOAP request authenticated as and retrieves the user_name then returns it to the SOAP client. The fakeOutTradePriceRequest function could be expanded to perform useful activities, such as looking up a stock symbol and returning the last traded price.
**generateSoapFault function**

The `generateSoapFault` function returns a SOAP error that can be called if there are problems.

Use the static WSDL

Load the static WSDL into a SOAP client to make requests to the SOAP web service.

The web service client provides

- The FakeStockValue project.
- The StockQuoteBinding web service.
- The GetLastTradePrice SOAP function. This function generates request records when run.

![FakeStockValue](image)

### Figure 823: Loaded WSDL

You can change the default request XML in the static WSDL to include a stock symbol.

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:stoc="http://example.com/stockquote.xsd">
  <soapenv:Header/>
  <soapenv:Body>
    <stoc:TradePriceRequest>IBM</stoc:TradePriceRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Submitting a SOAP request to this web service endpoint returns the following to the requesting SOAP client.

```xml
  <SOAP-ENV:Body>
    <GetLastTradePriceOutput xmlns="https://www.service-now.com/vws/FakeStockValue">
      <message>admin2, You were looking for a quote on IBM</message>
    </GetLastTradePriceOutput>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Inbound web service examples

Inbound web service examples demonstrate how to access ServiceNow web services.

#### Java Apache Axis2 web services client examples

Examples demonstrating an integration with Axis2 Version 1.4.
Requirements

- An "elementFormDefault" value of qualified means that an unqualified element is in the default namespace defined on an ancestor. If it is "unqualified" then an unqualified element is in the empty namespace (xmlns=""). The default is "unqualified".
- To Resolve the Axis Client deserialization failure you should go to System Properties > Web Services and uncheck the property that sets the elementFormDefault attribute of the embedded XML schema to the value of unqualified. Save the property setting and regenerate your Axis2 client code if your client code was generated before changing this property.

````java
public class Insert {
    public static void main ( String args [ ] ) { try {
        HttpTransportProperties. Authenticator basicAuthentication = new HttpTransportProperties. Authenticator ( ) ;
        basicAuthentication. setUsername ( "admin" ) ;
        basicAuthentication. setPassword ( "admin" ) ;

        ServiceNowStub proxy = new ServiceNowStub ( ) ;
        proxy._getServiceClient ( ). getOptions ( ). setProperty (org. apache. axis2. transport. http. HTTPConstants. AUTHENTICATE,
        basicAuthentication ) ;

        ServiceNowStub. Insert inc = new ServiceNowStub. Insert ( ) ;
        ServiceNowStub. InsertResponse resp = new ServiceNowStub. InsertResponse ( ) ;

        inc. setAssigned_to ( "Christen Mitchell" ) ;
        inc. setCategory ( "hardware" ) ;
        inc. setPriority ( BigInteger. ONE ) ;
        inc. setDescription ( "The WI_FI in the reception area is down" ) ;
        inc. setCaller_id ( "Joe Employee" ) ;

        resp = proxy. insert (inc ) ;

        System. out. println ( "New Incident: " + resp. getNumber ( ) ) ;
    } catch ( Exception e ) { System. out. println (e. toString ( ) ) ;}
```
Java Apache Axis2 web services client examples update
An example of an Axis Client program that calls the getKeys function to query all incidents where the category is Hardware.

getKeys

A list of sys_id is returned as a result:

```java
package com.service_now.www;

public class DemoClient {
    public static void main(String args[]) { try {
        ServiceNowStub proxy = new ServiceNowStub();
        ServiceNowStub.GetKeys getInc = new ServiceNowStub.GetKeys();
        ServiceNowStub.GetKeysResponse resp = new ServiceNowStub.GetKeysResponse();
        getInc.setActive(true);
        getInc.setCategory("hardware");
        proxy._getServiceClient().getOptions().setProperty(
            org.apache.axis2.transport.http.HTTPConstants.CHUNKED,
            Boolean.FALSE);
        resp = proxy.getKeys(getInc);
        String[] keys = resp.getSys_id();
        System.out.println("Key: " + keys[0]);
    } catch (Exception e) {
        System.out.println(e.toString());
    }
}
```

getRecords

```java
package com.service_now.www;

import com.service_now.www.ServiceNowStub.GetRecordsResult_type0;

public class GetRecords {
    /**
     * @param args
     */
    public static void main(String[] args) { try {
        ServiceNowStub proxy = new ServiceNowStub();
        ServiceNowStub.GetRecords incidents = new ServiceNowStub.GetRecords();
        incidents.setActive(true);
        incidents.setCategory("hardware");
        incidents.setSys_created_on("> 2009-06-08 10:30:00");
```
proxy._getServiceClient ( ). getOptions ( ). setProperty (org.apache.axis2.transport.http.HTTPConstants.CHUNKED, Boolean.FALSE);

result = proxy.getRecords (incidents);
GetRecordsResult_type0 [ ] keys = result.getGetRecordsResult ( );

for ( int key = 0 ; key < keys.length ; key ++ ) {
    System.out.println ( "Key: " + keys[0].getSys_id ( ) ) ;
} } catch ( Exception e ) {
    System.out.println (e.toString ( ) ) ;
}

Java Apache Axis2 web services client examples advanced
Examples showing how to construct and use an Axis2 client to consume a ServiceNow Web Service.

Axis is essentially a SOAP engine -- a framework for constructing SOAP processors such as clients, servers, or gateways. The current version of Axis is written in Java. This content is intended for system admins with a light development background in Java. To begin you would need Java JDK version 1.4.2 or higher and Axis2 version 1.0 or higher.

Create a Java Project

This example uses Eclipse SDK Version: 3.4.2 for managing the source code and executing the web request. Eclipse is not required.

• Open Eclipse and from the menu select File > New > Project > Java Project.
• Give the project a name.
• Verify that the correct JRE is specified.
  • If using wsdl2java run "java -version" on the command line and this will be the version to specify for the project specific JRE.
  • If using the Axis2 Codegen plugin use default JRE.
Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: TestWebService

Contents

- Create new project in workspace
- Create project from existing source

Directory: /glide/workspace/TestWebService

JRE

- Use default JRE (Currently 'JVM 1.4.2 (MacOS X Default)')
- Use a project specific JRE: JVM 1.6.0
- Use an execution environment JRE: J2SE-1.4

Project layout

- Use project folder as root for sources and class files
- Create separate folders for sources and class files

Working sets

- Add project to working sets

Working sets: 

Finish
Generate your Axis2 client code

- From a command line in the bin directory of the axis folder:

  ```
  ./wsdl2java.sh -uri https://<instance name>.service-now.com/incident.do?WSDL -o /glide/workspace/TestWebService/
  ```

- In the above example:
  - The "-uri" is either the path where you have saved a copy of the wsdl to either ".wsdl" or ".xml", or the URL the WSDL resides at.
  - The "-o" is the path where you want the files to be written out to. If not specified, the files will be written out to the current bin location.

- In Eclipse refresh the project and the generated Stub and CallbackHandler should now be displayed

  ![Figure 826: Axis Stub](image)

Basic Authentication

```java
HttpTransportProperties.Authenticator basicAuthentication = new HttpTransportProperties.Authenticator();
basicAuthentication.setUsername("admin");
basicAuthentication.setPassword("admin");
...
ServiceNowStub proxy = new ServiceNowStub();
...
proxy._getServiceClient().getOptions().setProperty(org.apache.axis2.transport.http.HTTPConstants.AUTHENTICATE, basicAuthentication);
```

Compatibility with Axis2 Versions 1.1 and higher

Chunking support is only available in HTTP Version 1.1. By default chunking is enabled in Axis2.xml for versions 1.1 and higher. ServiceNow does not support Chunking, so you will need to disable chunking at deployment time or at runtime.
• Deployment time: One can disable HTTP chunking by removing or commenting out the following element from Axis2.xml

```xml
<parameter name= "Transfer-Encoding" >chunked</parameter>
```

• Runtime: User can disable the chunking using following property set in Client or Stub, versions 1.1.1 and higher only

```java
options.setProperty (org.apache.axis2.transport.http.HTTPConstants.CHUNKED, Boolean.FALSE);
```

### Creating Unique Packages

You can use the Axis2 parameter namespace2package (ns2p) to create unique package names. The parameter uses this format:

```bash
<Axis path>\bin\wsdl2java.bat -u -p cr2 -ns2p <namespace>=<package name> -uri <wsdl to convert>
```

For example:

```bash
<Axis path>\bin\wsdl2java.bat -u -p cr2 -ns2p http://www.service-now.com/change_request=my.change_request -uri change_request
```

### Microsoft .NET web services client examples

Examples demonstrating an integration with Microsoft .NET Web Services Client.

### Requirements

.NET 2.0 Versions and Higher:

- An "elementFormDefault" value of qualified means that an unqualified element is in the default namespace defined on an ancestor. If it is "unqualified" then an unqualified element is in the empty namespace (xmlns=""), The default is "unqualified".
- To Resolve the .NET Client deserialization failure you should go to System Properties > Web Services and uncheck the property that sets the elementFormDefault attribute of the embedded XML schema to the value of unqualified. Save the property setting and recreate your WSDL Reference.cs class. See Also "Compatibility with Clients generated from WSDL" below.
Figure 827: Element Form Default Property

Example insert using Visual Basic .NET
A sample Visual Basic .NET program that inserts a core_company record.

```vbnet
Public Class Class1
    Shared Sub Main()
        Dim proxyCompany As New core_company.localhost.ServiceNow()

        Dim companyInsert As New core_company.localhost.insert()

        Dim companyInsertResponse As core_company.localhost.insertResponse

        With companyInsert
            .name = "Test Company SKF 2"
            .contact = "SKF"
            .customer = True
            .customerSpecified = True
        End With

        companyInsertResponse = proxyCompany.insert(companyInsert)

        Console.WriteLine(companyInsertResponse.sys_id)
    End Sub End Class
```

Sample Visual Basic .NET project
Perl web services client examples
Examples demonstrating an integration with a Perl web services client.

Note: The following examples require the usage of the Perl language and the SOAP::Lite package.

System Requirements

- Perl 5.8
  - SOAP::Lite (prerequisites http://soaplite.com/prereqs.html)
  - Crypt::SSLeay
  - IO::Socket::SSL

insert

The following example will insert a record into the Incident table.

```perl
#!/usr/bin/perl -w
#
# declare usage of SOAP::Lite
use SOAP::Lite;
#
# specifying this subroutine, causes basic auth to use# its credentials when# challenged
sub SOAP::Transport::HTTP::Client::get_basic_credentials{
  # login
  as the itil userreturn 'itil' =>'itil';
}
#
# declare the SOAP endpoint here
my $soap= SOAP::Lite->proxy('https://myinstance.service-now.com/incident.do?SOAP');
#
# calling the insert function
my $method= SOAP::Data->name('insert')-
attr({xmlns =>'http://www.service-now.com/'});
#
# create a new incident with the following short_description and
# category
my @params=( SOAP::Data->name(short_description =>'This is an
# example short description'));push @params, SOAP::Data->name(category
# =>'Hardware');
#
# invoke the SOAP call
my $result=$soap->call($method=@params);
#
# print any SOAP faults that get returned
print_fault($result);# print the SOAP response that get return
print_result($result);
#
# convenient subroutine for printing all resultssub print_result
  (my $result)=@_
  if($result->body&&$result->body->{'insertResponse'}){
    my %keyHash=%{$result->body->{'insertResponse'}};
    foreach my $k(keys %keyHash){print "name=$k   value=
    $keyHash{$k}\n";}
  }
#
# convenient subroutine for printing all SOAP faultssub print_fault
  (my $result)=@_
  if($result->fault){print "faultcode=".$result->fault-
# ('faultcode')."\n";print "faultstring=".$result->fault-
# ('faultstring')."\n";print "detail=".$result->fault->('detail')."\n";}
```

© 2017 ServiceNow. All rights reserved. 3363
**insert (With XML payload)**

The following is an example of inserting a record into the ecc_queue table where the payload field is an XML document. This is done using the Perl language and the SOAP::Lite package, the XML document creation uses the XML::Writer package:

```perl
#!/usr/bin/perl -w
use SOAP::Lite ( +trace => all, maptype => {} );
use SOAP::Lite;
use XML::Writer;
use XML::Writer::String;

## Get parameters passed by OVO notification
$OVMSG{id}=$ARGV[0];
$OVMSG{node_name}=$ARGV[1];
$OVMSG{node_type}=$ARGV[2];
$OVMSG{date_created}=$ARGV[3];
$OVMSG{time_created}=$ARGV[4];
$OVMSG{date_received}=$ARGV[5];
$OVMSG{time_received}=$ARGV[6];
$OVMSG{application}=$ARGV[7];
$OVMSG{msg_group}=$ARGV[8];
$OVMSG{object}=$ARGV[9];
$OVMSG{severity}=$ARGV[10];
$OVMSG{operator_list}=$ARGV[11];
$OVMSG{msg_text}=$ARGV[12];
$OVMSG{instruction}=$ARGV[13];

sub SOAP::Transport::HTTP::Client::get_basic_credentials{return 'itil' =>'itil';}

my $soap = SOAP::Lite->proxy('http://<instance name>.service-now.com/ecc_queue.do?SOAP');

my $method = SOAP::Data->name('insert') ->attr({xmlns => 'http://www.service-now.com/'});

# get all incidents with category Network
my @params = (SOAP::Data->name(agent =>'OVO_Notification'));
push(@params, SOAP::Data->name(queue =>'input'));
push(@params, SOAP::Data->name(name =>'HP Openview OVO Notification'));
push(@params, SOAP::Data->name(source =>$OVMSG{id}));

my $s = XML::Writer::String->new();
my $writer = new XML::Writer(OUTPUT => $s);

#$writer->xmlDecl();$writer->startTag('notification');

write_element('id');
write_element('node_name');
write_element('node_type');
write_element('date_created');
write_element('time_created');
write_element('date_received');
write_element('time_received');
write_element('application');
write_element('msg_group');
write_element('object');
write_element('severity');
write_element('operator_list');
write_element('msg_text');
write_element('instruction');

$writer->endTag('notification');

$writer->end;

sub write_element {my $label=shift;my $value=$OVMSG{$label};$writer->startTag($label);if($value){$writer->characters($value);$writer->endTag($label);}}
push(@params, SOAP::Data->name(payload =>$s->value()));

print $soap->call($method=>@params)->result;
```
=== Response to the ''insert''===

```xml
<?xml version="1.0" encoding="UTF-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <insertResponse xmlns="http://www.service-now.com/ecc_queue">
      <sys_id>1a5ad50e0a0a021101bef2e07705f87a</sys_id>
      <name>HP Openview OVO Notification</name>
    </insertResponse>
  </soap:Body>
</soap:Envelope>
```

**update**

```perl
#!/usr/bin/perl -w
use SOAP::Lite (?trace => all, maptype => {}); use SOAP::Lite;
sub SOAP::Transport::HTTP::Client::get_basic_credentials { return 'itil' => 'itil'; }
my $soap = SOAP::Lite->proxy('http://localhost:8080/glide/incident.do?SOAP');
my $method = SOAP::Data->name('update')->attr({xmlns => 'http://www.service-now.com/'});
# update incident by sys_id
my @params = ( SOAP::Data->name(sys_id => 'e8caedcbc0a80164017df472f93e9ed1'));
push(@params, SOAP::Data->name(short_description => 'this is a new description'));
my $result = $soap->call($method => @params);
print_fault($result);
print_result($result);
sub print_result { my($result) = @_;
  if($result->body && $result->body->{'updateResponse'} ) {
    my %keyHash = $result->body->{'updateResponse'};
    foreach my $k (keys %keyHash) {
      print "name=$k   value=$keyHash{$k}\n";
    }
  }
}
sub print_fault { my($result) = @_;
  if($result->fault) {
    print "faultcode=".$result->fault->{'faultcode'}."\n"
    print "faultstring=".$result->fault->{'faultstring'}."\n"
    print "detail=".$result->fault->{'detail'}."\n";
  }
}
```

**getKeys**

The following is an example of retrieving a list of keys for records of Incident where status is Network.

```perl
#!/usr/bin/perl -w
use SOAP::Lite (?trace => all, maptype => {}); use SOAP::Lite;
sub SOAP::Transport::HTTP::Client::get_basic_credentials { return 'itil' => 'itil'; }
my $soap = SOAP::Lite->proxy('http://<instance name>.service-now.com/incident.do?SOAP');
```
my$method= SOAP::Data->name('getKeys')->attr({xmlns =>'http://www.service-
    now.com/'});

# get all incidents with category Networkmy@params=( SOAP::Data-
    ->name(category =>'Network'));

print$soap->call($method=>@params)->result;

get

The following is an example of retrieving an Incident record using its sys_id value

#!/usr/bin/perl -w#use SOAP::Lite ( +trace => all, maptype => {} );use SOAP::Lite;

sub
    SOAP::Transport::HTTP::Client::get_basic_credentials{return 'itil' =>'itil';}

my$soap= SOAP::Lite->proxy('http://<instance name>.service-now.com/
    incident.do?SOAP');

my$method= SOAP::Data->name('get')->attr({xmlns =>'http://www.service-
    now.com/'});

# get incident by sys_idmy@params=( SOAP::Data->name(sys_id
    =>'9d385017c611228701d22104cc95c371'));

my%keyHash=%{$soap->call($method=>@params)->body->{'getResponse'}};

# iterate through all fields and print themforeach my$k(keys%keyHash)
    {print "$k=$keyHash{$k}\n";}

getRecords

To query for an Incident using its incident number value:

#!/usr/bin/perl -w#use SOAP::Lite ( +trace => all, maptype => {} );use SOAP::Lite;

sub
    SOAP::Transport::HTTP::Client::get_basic_credentials{return 'itil' =>'itil';}

my$soap= SOAP::Lite->proxy('http://<instance name>.service-now.com/
    incident.do?SOAP');

my$method= SOAP::Data->name('getRecords')->attr({xmlns =>'http://
    www.service-now.com/'});# get incident by numbermy@params=( SOAP::Data-
    ->name(number =>'INC10001'));

my%keyHash=%{$soap->call($method=>@params)->body->{'getRecordsResponse'}-
    >{'getRecordsResult'});

# iterate through all fields and print themforeach my$k(keys%keyHash)
    {print "$k=$keyHash{$k}\n";}

© 2017 ServiceNow. All rights reserved. 3366
getRecords (Returning Multiple Results)

The following is an example of retrieving and displaying an array of Incident records by querying all incidents that have a category of "Network".

```perl
#!/usr/bin/perl -w
use SOAP::Lite (+trace => all, maptype => {});
sub SOAP::Transport::HTTP::Client::get_basic_credentials { return 'itil' => 'itil'; }
my $soap = SOAP::Lite->proxy('http://<instance name>.service-now.com/incident.do?SOAP');
my $method = SOAP::Data->name('getRecords')->attr({xmlns => 'http://www.service-now.com/'});
# get incident by sys_id
my @params = (SOAP::Data->name(category => 'Network'));
my %keyHash = %{$soap->call($method => @params)->body->{'getRecordsResponse'}};
my $i = 0; my @size = @{$keyHash{'getRecordsResult'}}; for ($i = 0; $i < @size; $i++) { my %record = @{$keyHash{'getRecordsResult'}}[$i]; print "------------------------------ $i ----------------------------
";
foreach my $kk (keys %record) { print "$kk = $record{$kk} \n"; }
```

deleteRecord

```perl
#!/usr/bin/perl -w
use SOAP::Lite (+trace => all, maptype => {});
sub SOAP::Transport::HTTP::Client::get_basic_credentials { return 'itil' => 'itil'; }
my $soap = SOAP::Lite->proxy('http://localhost:8080/glide/incident.do?SOAP');
my $method = SOAP::Data->name('deleteRecord')->attr({xmlns => 'http://www.service-now.com/'});
# delete incident by sys_id
my @params = (SOAP::Data->name(sys_id => '46f67787a9fe198101e06dfcf3a78e99'));
my $result = $soap->call($method => @params);
print_fault($result);
print_result($result);
sub print_result { my ($result) = @_; if ($result->body && $result->body->{'deleteRecordResponse'}) { my %keyHash = %{$result->body->{'deleteRecordResponse'}};
foreach my $k (keys %keyHash) {
print "name = $k   value = $keyHash{$k} \n"; }
}
sub print_fault { my ($result) = @_; }
```
Python web services client examples
Examples demonstrating an integration with a Python web services client.

Requirements
The following examples require the installation of the following Python modules:

- fpconst [http://pypi.python.org/pypi/fpconst/0.7.2](http://pypi.python.org/pypi/fpconst/0.7.2)
- PyXML [http://pyxml.sourceforge.net/topics/](http://pyxml.sourceforge.net/topics/)

insert
This is an example of inserting an incident.

```python
#!/usr/bin/python
from SOAPpy import SOAPProxy
import sys

def createincident (params_dict):
    # instance to send to
    instance = 'demo'

    # username/password
    username = 'itil'
    password = 'itil'

    # proxy - NOTE: ALWAYS use https://INSTANCE.service-now.com, not https://www.service-now.com/INSTANCE for web services URL from now on!
    proxy = 'https://%s:%s@%s.service-now.com/incident.do?SOAP' % (username, password, instance)
    namespace = 'http://www.service-now.com/
    server = SOAPProxy (proxy, namespace)

    # uncomment these for LOTS of debugging output
    #server.config.dumpHeadersIn = 1 #server.config.dumpHeadersOut = 1
    #server.config.dumpSOAPOut = 1 #server.config.dumpSOAPIn = 1

    response = server. insert (impact = int (params_dict ['impact']) , urgency = int (params_dict ['urgency']) , priority = int (params_dict ['priority']) , category =params_dict ['category'] , location =params_dict ['location'] , caller_id =params_dict ['user'] , assignment_group =params_dict ['assignment_group'] , assigned_to =params_dict ['assigned_to'] , short_description =params_dict ['short_description'] , comments =params_dict ['comments'] )

    return response

values = { 'impact': '1', 'urgency': '1', 'priority': '1', 'category': 'High', 'location': 'San Diego', 'user':
```
'fred.luddy@yourcompany.com', 'assignment_group': 'Technical Support', 'assigned_to': 'David Loo', 'short_description': 'An incident created using python, SOAPpy, and web services.', 'comments': 'This a test making an incident with python. Isn't life wonderful?'}

new_incident_sysid = createincident(values)
print "Returned sysid: " + repr(new_incident_sysid)

getKeys

This is an example of executing getKeys on the demo instance using basic authentication.

#!/bin/env python
# use the SOAPpy module from SOAPpy import SOAPProxy

username, password, instance = 'admin', 'admin', 'demo'

server = SOAPProxy(proxy, namespace)
response = server.getKeys(category = 'Network')
print response.sys_id.split(',')

getRecords

In this example, we get an incident, querying for category == "Network" (with basic authentication).

#!/bin/env python
# use the SOAPpy module from SOAPpy import SOAPProxy

username, password, instance = 'admin', 'admin', 'demo'

server = SOAPProxy(proxy, namespace)
response = server.getRecords(category = 'Network')

for record in response:
    for item in record:
        print item

get

In this example, we get an incident record by sys_id (with basic authentication).

#!/bin/env python
# use the SOAPpy module from SOAPpy import SOAPProxy

username, password, instance = 'admin', 'admin', 'demo'
Advanced

This is an example of advanced Python script that reads a log file for a keyword invalid spi and creates an ECC Queue record where the payload is set to an alert of XML format.

```python
#!/bin/env python

# kevin.pickard@service-now.com  2008.07.03  initial creation

from SOAPpy import SOAPProxy  
from xml.dom.minidom import getDOMImplementation
import sys, os, socket, pickle, re

# instance to send to
instance = 'demo'

# username/pass
username = 'admin'
password = 'admin'

# log file to watch
syslogfile = '/var/log/cisco.log.ksp'

# state file
statefile = '/tmp/syslog_ecc.state-test'

# ECC queue values
soapagent = 'SOAPpy'
ecctopic = 'PIX Error: '
ecccname = 'Invalid SPI: '
eccsource = 'Syslog'

# regex string to match
matchstring = 'invalid spi'

try:
    state = open (statefile , 'r' )
    lastbyte = pickle.load (state )
    state. close () except:
    lastbyte = 0

    #print 'DEBUG: lastbyte = '+str(lastbyte)

try:
    log = open (syslogfile , 'ro' ) except:
    errortopic = 'Script Error'
    errorname = 'Unable to open log file '+syslogfile+ '.
    errorpayload = 'This message was generated due to an error condition encountered in a script. The name of the script is '+ os. path. basename ( sys. argv [ 0 ] )+ ' on server '+ socket. gethostname ()+ ' .'  

    proxy = 'https://'+username+ ':'+password+ '@'+instance+ '.service-now.com/ecc_queue.do?SOAP'  
    namespace = 'http://www.service-now.com/'
```
server = SOAPProxy (proxy, namespace)
server.config.dumpSOAPOut = 1
server.config.dumpSOAPIn = 1
response = server.insert (agent=soapagent, topic=errortopic, name=errorname, source=sys.argv[0], payload=errorpayload)
sys.exit(1)

if lastbyte != 0:
    try:
        log.seek(lastbyte) except IOError:
            pass

loglines = log.readlines()
lastbyte = log.tell()
log.close()

state = open(statefile, 'w')
pickle.dump(lastbyte, state)
state.close()

# regex out the line
matchedlines = []
for line in loglines:
    if re.search(matchstring, line) != None:
        matchedlines.append(line)

if len(matchedlines) == 0:
    sys.exit(0)

proxy = 'https://' + username + ':' + password + '@' + instance + '.service-now.com/

namespace = 'http://www.service-now.com/'

server = SOAPProxy (proxy, namespace)  # server.config.dumpSOAPOut = 1
# server.config.dumpSOAPIn = 1

entriestosend = {}
for line in matchedlines:
    device = line.split()[3]
    sourceip = line.split()[-1]
    entriestosend[sourceip] = [device, line]

for key, value in entriestosend.iteritems():
    impl = getDOMImplementation()
    newdoc = impl.createDocument(None, "log_line", None)
    top_element = newdoc.documentElement
    text = newdoc.createTextNode(value[1])
    top_element.appendChild(text)

response = server.insert (agent=soapagent, topic=ecctopic+value[0], name=eccname+key, source=eccsource, payload=value[1])

Web services C Sharp .NET end to end tutorial

Examples demonstrating how to use .NET to consume a ServiceNow web service.

This tutorial will show you how to configure ServiceNow correctly to receive a web service request from your .NET client, as well as how to consume our web services using C# .NET.
Configure C# with .NET
Configure web services within ServiceNow.

1. To configure web services within ServiceNow, access the **System Properties > Web Services** module.

   This module displays the system properties that are specific to web services within your instance. For security reasons, you will want to make sure that you require basic authorization for incoming SOAP requests. This ensures that only authenticated users will be able to make any web services calls, whether it be via web service import sets or inserting/deleting/querying via direct web services.

![Web Services](image)

   **Please edit your changes and press Save**

   **Customization Properties for Web Services**

   **Require basic authorization for incoming RSS requests**
   - [ ] Yes | No

   **Require basic authorization for incoming SOAP requests**
   - [ ] Yes | No

2. This next step is very important if you are using .NET as a client to connect to ServiceNow. You must set the `elementFormDefault` property to false.

   This property defines how the WSDLs are qualified. Of course, if you do not consume our WSDL and just create the XML manually, then this property is irrelevant.

   `[For further documentation, follow this URL: http://wiki.servicenow.com/index.php?title=Web_Services]`
   - [ ] Yes | No

---

*Call a web service in visual studio .NET*

Call a web service using Visual Studio 2008.

In this example, we will be using Visual Studio 2008. First, create a new Windows Form Application for this example.
Figure 829: Dot net project

On the resulting form, we created a richTextBox (which we named 'richTextBoxResult') and a button (named buttonResult).
Use a service reference in a C Sharp integration

Use a wizard to add a service reference for a C Sharp integration.

Go to the Solutions Explorer and select Service References > Add Service Reference. A wizard will appear asking for an address. Use: https://<instance name>.service-now.com/incident.do?WSDL. Accept the defaults for the rest of the wizard.

Open the app.config file and change the Security mode to “Transport” and the clientCredentialType and proxyCredentialType to “Basic”
Use a web reference in a C Sharp integration

Use a wizard to add a web reference for a C Sharp integration.

Go to the Solutions Explorer and select Service References > Add Service Reference. A wizard will appear. At the bottom of the form, there is an Advanced button. Click on it and click on the Add Web Reference button at the bottom of the new wizard page. This will start the Web Reference wizard. For the URL, use: https://<instance name>.service-now.com/incident.do?WSDL and name the web reference, 'WebReference1'. Accept the defaults for the rest of the wizard.

C Sharp integration source code

After defining the source code, insert it.

Now we are ready to insert the code. Double-click on the Send Web Service button on your form to open the backend code to the form that has been created. Here is the code to insert a record into the demo instance and to read the response.

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ExampleWebServiceForWiki
{
    public partial class FormMain : Form
    {
        public FormMain()
        {
            InitializeComponent();
        }

        private void buttonSend_Click(object sender, EventArgs e)
        {
            /* SERVICE REFERENCE-SPECIFIC CODE
             ServiceReference1.ServiceNowSoapClient soapClient = new ServiceReference1.ServiceNowSoapClient();
             soapClient.ClientCredentials.UserName.UserName = "itil";
             soapClient.ClientCredentials.UserName.Password = "itil";
             ServiceReference1.insert insert = new ExampleWebServiceForWiki.ServiceReference1.insert();
             ServiceReference1.insertResponse response = new ExampleWebServiceForWiki.ServiceReference1.insertResponse();
             // END OF SERVICE REFERENCE CODE */
```
C Sharp integration results
If you have followed the tutorial correctly, you should receive the result whether you used a Service Reference or a Web Reference.

If you have followed this tutorial correctly, you should receive the following result whether you used a Service Reference or a Web Reference.

Figure 832: Dot net tutorial results
Troubleshoot a null response in a C Sharp integration
Receiving a null response from ServiceNow's web service.

If you are receiving a "null" response from your web service in your client code, then you may have missed the step in this tutorial for setting the elementFormDefault setting to "False". Here is a quick video tutorial that shows you how to do this.

Remember to recompile your code against the WSDL after you have changed this setting and saved it.

Retrieve a large number of records using SOAP
By default, a single SOAP request can retrieve a maximum of 250 records.

SOAP relies on Extensible Markup Language (XML) as its message format, and usually relies on other Application Layer protocols (most notably Remote Procedure Call (RPC) and HTTP) for message negotiation and transmission. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built.

Because of the verbose XML format, SOAP can be considerably slower than other transport methods. Therefore, sending a large amount of data via SOAP is inefficient and is discouraged. Because of this, ServiceNow has imposed a hard-limit of 250 records that can be retrieved at any time in a single query. You may find that this limit poses some technological challenges for your integration design.

**SOAP strategies**
Retrieve the information that you need and make your integration more efficient.

Use filters to limit the number of results
One of the best ways to make your web service calls fit within the 250 record limit is to think about the design of your integrating application.

For example, let's assume that we are making an incident form in C# to show a user the incidents that are assigned to him.

**Problematic Approach**
The C# application makes a soap call to retrieve all of the incidents within ServiceNow. The application would then store the results locally in memory. When the user decides to view the incidents that are assigned to him, the application loops the the internal array and displays the incidents that are assigned to the user.
**A Better Approach**

The C# application makes a soap call to retrieve all of the incidents within ServiceNow that are assigned to the logged-in user. The results are stored locally in memory. When the user decides to view the incidents that are assigned to him, the application shows all the results to the user.

**The Best Practice**

The C# application makes no SOAP call initially. When a logged-in user decides to view the incidents that are assigned to him, the application presents him with the choice of viewing active, closed, etc. It gives him the ability to filter the results that he wants to see before the SOAP call is even made. Then, the user is only presented with the results that he wished to view.

Use a local data store to pull data from

If a large amount of data needs to be queried often, and the data does not need to be real-time, perform a sync of the ServiceNow table that you're interested in with your integrating application's data store.

**Data push**

- Using a scheduled job, ServiceNow can generate a csv/xml from a report and have it emailed to a specific location. The receiver might have a trigger to take the email attachment, parse it, and populate an internal table from which the application can communicate when the data is needed.
- Using a schedule job, ServiceNow can generate a csv/xml from a report and FTP it to an public FTP/FTPS location. The integrating product would consume this csv file on a regular basis and populate an internal table from which the application can communicate when the data is needed.

**Note:** Currently, the platform does not provide a method for extracting very large amounts of data and sending the output to an FTP server. However, a customization to perform that function is described at [here](#). The customization was developed for use in specific ServiceNow instances, and is not supported by ServiceNow Customer support. The method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

**Data pull**

Using a cron job, a machine internal to your network can make a wget call to pull csv/xml data from any table within ServiceNow. The integrating product would consume this csv/xml file on a regular basis and populate an internal table from which the application can communicate when the data is needed. Examples of the wget command that would be used:

- wget --user=itil --password=itil --no-check-certificate https://<instance name>.service-now.com/incident_list.do?CSV
- wget --user=itil --password=itil --no-check-certificate https://<instance name>.service-now.com/incident_list.do?XML

Use Java/C#/PHP code to fetch the XML data using basic authentication

If a local data store is not an option, another way to get the data is to call the CSV/XML processor directly and then parse the results.

Use the resulting data in a similar manner as you would a direct SOAP call. An example of this in PHP:
//This example is in PHP

$user = "itil";
$pass = "itil";
$userPass = $user.':'.$pass;

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://<instance name>.service-now.com/incident_list.do?CSV');
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_USERPWD, $userPass);
$data = curl_exec($ch);
$info = curl_getinfo($ch);

if ($output === false) {
    $output = "No cURL data returned for $addr ["." $info['http_code']]. ""]";
    if (curl_error($ch))
        $output .= "\n". curl_error($ch);
    print $output;
} else{
    echo $data;
} 
curl_close($ch);

EXCEL web service

The platform supports programmatic retrieval of Excel binary formatted data through an HTTP GET request.

The request is triggered by use of a URL parameter EXCEL. For example, to retrieve a list of incidents and save them to an Excel file, issue an HTTP GET to the following URL:

https://instance_name.service-now.com/incident.do?EXCEL

The Content-Disposition HTTP header in the response displays the file name and extension of the resulting file. In the example, it will be "incident.xls".

EXCEL web service parameters

Use URL parameters to customize and filter the response.

**sysparm_query**

Enter an encoded query string in this parameter. The parameter filters the data using the encoded query before returning the Excel file. The following request filters the list to return only incidents that are active:

https://instance_name.service-now.com/incident.do?
EXCEL&sysparm_query=active=true
**sysparm_view**

The value of this URL parameter indicates which list view to use to limit the field values that are returned. For example, to use the "ess" view:

```
https://instance_name.service-now.com/incident.do?EXCEL&sysparm_view=ess
```

**CSV Web service**

An administrator can retrieve CSV data using an HTTP GET request.

Trigger a request using a CSV URL parameter. For example, to retrieve a list of incidents and save them to a CSV file, issue an HTTP GET to the following URL: `https://instance_name.service-now.com/incident.do?CSV`

The Content-Disposition HTTP header in the response indicates the file name and extension of the extract; "incident.csv" in the example.

**CSV Web service parameters**

The following URL parameters customize and filter the CSV content that is returned by an HTTP GET request.

**sysparm_query**

Enter an encoded query string in this parameter. Use the parameter to filter the data using the encoded query before returning the CSV content. The following request filters the list to return only active incidents:

```
https://instance_name.service-now.com/incident.do?CSV&sysparm_query=active=true
```

**sysparm_view**

The value indicates which view to use to limit the field values that are returned. For example, to use the "ess" view:

```
https://instance_name.service-now.com/incident.do?CSV&sysparm_view=ess
```

**Adding data to tables by posting a CSV file**

You can create new data by posting a CSV file directly to a table.

The CSV file headers must match the field columns in the targeted table.

**JSONv2 Web Service**

The ability to describe sets of data in JSON format is a natural extension to the JavaScript language. ServiceNow supports a web service interface that operates on the JSON object as the data input and output format.

The JSONv2 web service is provided by a platform-level processor similar to the services for SOAP, WSDL, CSV, Excel, and XML. Like those services, the JSON web service is triggered by a standalone JSONv2 URL parameter. For example:

```
https://<instance name>.service-now.com/mytable.do?JSONv2
```
Having the JSON object available as a data format for web services means that you can create (insert), update, and query any data in the system using the JSON object format, and get results in the JSON object format.

**Security**

Like all other HTTP-based web services available on the platform, the JSONv2 web service is required to authenticate using basic authentication by default. The user ID that is used for authentication is subjected to access control in the same way as an interactive user.

**JSON object format**

The JSON object is built in two structures.

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

In its simplest form, a JSON object is just a comma delimited set of name/value pairs. For example:

```json
{"name one":"value one","name two":"value two"}
```

The following is a sample of a single record array of incidents in JSON:

```json
{"records": [
{"closed_by":null,
"__status": "success",
"category":"inquiry",
"escalation":"0",
"state":"1",
"location":null,
"reassignment_count":"0",
"time_worked":null,
"order":0,
"due_date":null,
"number":"INC0010180",
"upon_approval":"proceed",
"sla_due":"2010-03-04 22:51:49",
"follow_up":null,
"notify":1,
"business_stc":0,
"caused_by":null,
"rejection_goto":null,
"assignment_group":"d625dccecc0a8016700a222a0f7900d06",
"incident_state":1,
"opened_at":"2010-02-23 22:51:49",
"wf_activity":null,
"calendar_duration":null,
"group_list":null,
"caller_id":null,
"comments":null,
"priority":3,
"sys_id":"fd0774860a0a0b380061bab9094733ad",
"sys_updated_by":"itil",
"variables":null,
"delivery_task":null,
"sys_updated_on":"2010-02-23 22:51:49",
"parent":null,
"active":true,
"opened_by":"681b365ec0a80164000fb0b5854a0cd",
"closed_by":null,
"__status": "success",
"category":"inquiry",
"escalation":"0",
"state":1,
"location":null,
"reassignment_count":0,
"time_worked":null,
"order":0,
"due_date":null,
"number":null,
"upon_approval":null,
"sla_due":null",
"follow_up":null,
"notify":1,
"business_stc":null,
"caused_by":null,
"rejection_goto":null,
"assignment_group":null,
"incident_state":null,
"opened_at":null,
"wf_activity":null,
"calendar_duration":null,
"group_list":null,
"caller_id":null,
"comments":null,
"priority":null,
"sys_id":null,
"sys_updated_by":null,
"variables":null,
"delivery_task":null,
"sys_updated_on":null,
"parent":null,
"active":null,
"opened_by":null
]}
```
The following is a record array of incident responses with an error.

```json
{
  "records": [
    {
      "__error": {
        "message": "Invalid Insert into: incident",
        "reason": "Data Policy Exception: Short description is mandatory"
      },
      "__status": "failure",
      "active": "true",
      "activity_due": "",
      "approval": "not requested",
      "approval_history": "",
      "approval_set": "",
      "assigned_to": "",
      "assignment_group": "d625dccec0a8016700a222a0f7900d06",
      "business_duration": "",
      "business_stc": "",
```
```
JSON response status

JSONv2 requests may return one of several response statuses.

**JSON Success Response**

Each JSON success response includes a record array containing the records retrieved by the given action. Each JSON object contains one or more metadata elements, prefixed with __, regarding the status for the action on each record, as illustrated in the previous examples. The JSON success responses use the following syntax:

```
__status
"__status": "<value>"
```

where <value> is success or failure.

__error

When the `status` element returns failure, the `error` element is added to identify the error and reason.

```
"__error": { "message": "<error value>", "reason": "<reason value> " }
```

where `<error value>` is the error message text and `<reason value>` is the reason the error was triggered.

**JSON Failure Response**

Each JSON failure response contains only the error and reason elements, which indicate the error message and more details about the error. Generally, this indicates that the whole JSON operation failed and no records can be processed.

For example:

```
{"error":"Cannot update with empty sysparm_query","reason":null}
```

**Setting the number of rows returned**

The following system property controls how many rows JSON returns with each query.
Table 1086: Setting the Number of Rows Returned

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.processor.json.row_limit</td>
<td>Specify the maximum number of rows a JSON query returns.</td>
</tr>
<tr>
<td></td>
<td>• Type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 10,000</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Properties</td>
</tr>
</tbody>
</table>

Requiring basic authentication for incoming JSONv2 requests

The following system property controls whether basic authentication is required for incoming JSONv2 requests.

Table 1087: Requiring Basic Authentication for Incoming JSONv2 Requests

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.basicauth.required.jsonv2</td>
<td>Enables (true) or disables (false) requiring basic authentication for incoming JSONv2 requests.</td>
</tr>
<tr>
<td></td>
<td>• Type: true</td>
</tr>
<tr>
<td></td>
<td>• Default value: true</td>
</tr>
<tr>
<td></td>
<td>• Location: Add to the System Properties</td>
</tr>
</tbody>
</table>

Action parameters

Action parameters are separate and different from data parameters because they specify the action to take when the JSON object parameter is part of an HTTP GET or POST request.

The parameters can also be specified as a field in the supplied JSON object. They have the effect of triggering an action in the case of sysparm_action, or filtering the results of an update or query in the case of sysparm_query.

**sysparm_action**

The following are the valid values for sysparm_action and the corresponding action triggered by the API.

Table 1088: Data Retrieval

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getKeys</td>
<td>Query the targeted table using an encoded query string and return a comma delimited list of sys_id values.</td>
</tr>
<tr>
<td>getRecords</td>
<td>Query the targeted table using an encoded query string and return all matching records and their fields.</td>
</tr>
</tbody>
</table>
**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Query a single record from the targeted table by specifying the <code>sys_id</code> in the <code>sysparm_sys_id</code> URL parameter, and return the record and its fields.</td>
</tr>
</tbody>
</table>

**Table 1089: Data Modification**

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>insert</td>
<td>Create one or more new records for the table targeted in the URL.</td>
</tr>
<tr>
<td>insertMultiple</td>
<td>Create multiple new records for the table targeted in the URL.</td>
</tr>
<tr>
<td>update</td>
<td>Update existing records in the targeted table in the URL, filtered by an encoded query string.</td>
</tr>
<tr>
<td>deleteRecord</td>
<td>Delete a record from the table targeted in the URL by specifying its <code>sys_id</code> in the <code>sysparm_sys_id</code> URL parameter.</td>
</tr>
<tr>
<td>deleteMultiple</td>
<td>Delete multiple records from the table targeted in the URL, filtered by an encoded query string.</td>
</tr>
</tbody>
</table>

**sysparm_query**

Specify an encoded query string to be used in `get`, `getRecords`, `update` or `deleteMultiple` `sysparm_action` value.

**sysparm_view**

Specify a form view to customize the return values for `get` and `getRecords` function calls. When using a view, the query returns only the fields defined in the view, including referenced values. If there is no view name, or if the view name is not valid, then the query returns all field names that are marked active in the dictionary.

**sysparm_sys_id**

Specify a target `sys_id` during a `get` or `delete` function call (`sysparm_action` value).

**sysparm_record_count**

Specify an integer value to limit the number of records retrieved for this request. Note that this value is capped by the `glide.processor.json.row_limit` system property.
**displayvalue**

Get the display value of a reference field, if any are in the record. For example, the Incident record can have an `assigned_to` field that is a reference to a user record. Instead of sending the `sys_id` of the user record, the user name is sent.

The `displayvalue` parameter can have three values: `true`, `false`, or `all`.

- **true**: All the references fields show the display value instead of `sys_id`.
- **false** (default): All reference fields show `sys_ids`.
- **all**: The display value and the `sys_id` are shown. For example, the `assignedto` field in the Incident record is sent back as `assigned_to:1234556, dv_assigned_to:Fred Luddy`.

**displayvariables**

Set this boolean value to **true** during a get or getRecords function call to retrieve all variables attached to this record.

**Data retrieval API**

Query for data by issuing an HTTPS GET request to the instance.

By default, a GET request is interpreted as a get function if a `sysparm_sys_id` parameter is present. Otherwise, it is interpreted as a getRecords function. You can also specify a URL parameter `sysparm_action=get`. Query responses are always encapsulated by a records hash of records, where each individual record's values are themselves hashed by field name.

**Return Display Value for Reference Variables**

When you are getting a record from a `get` or `getRecords` function, all the fields associated with that record are returned. The fields are often reference fields that contain a `sys_id` for another table. The base system behavior is to return the `sys_id` value for those fields. To have the display value for the field returned, use one of these options:

- Add the property `<tt>glide.json.return_displayValue</tt>` to the system properties, and every JSON request will return a display value for a reference field.
- Add the parameter `displayvalue=true` to the JSON request URL and JSON requests with that parameter will return a display value instead of the `sys_id` for a reference field. The JSON URL would look like this:

  ```
  https://<instance name>.service-now.com/incident.do?
  JSON&sysparm_action=getRecords&sysparm_query=active=true^category=hardware&displayvalue=true
  ```

- Add the parameter `displayvalue=all` to the JSON request URL and JSON requests with that parameter return a display value and the `sys_id` for a reference field. The response element name for the display value field will be prefixed with `dv_`, for example `dv_caller_id`.

**Return Display Variables**

Use the `display variables` parameter to return an array of variables associated with a service catalog item record when using a `get` or `getRecords` function. The variables are expressed hierarchically. A variable container has the container's variables in its children field.
To get display variables, add the parameter displayvariables=true to the JSON request URL. The JSON request URL would look like:

```plaintext
https://<instance name>.service-now.com/sc_req_item.do?
JSONv2&sysparm_action=getRecords&
sysparm_query=active=true^short_description=Laptop%20preconfigured%20for%20developers&displayvariables=true
```

White space has been added to the following example for clarity:

```json
```

**Control the order of records**

You can control the order that records appear in the JSON response. To set an order, use the ORDERBY or ORDERBYDESC clauses in the URL encoded query. For example,

```plaintext
sysparm_query=active=true^ORDERBYnumber^ORDERBYDESCcategory
```

filters all active records and orders the results in ascending order by number first, and then in descending order by category. For more information, see Encoded query strings.

**getKeys**

Get the sys_id of multiple records by specifying an encoded query string in the sysparm_query parameter.

```plaintext
https://<instance name>.service-now.com/incident.do?
JSONv2&sysparm_action=getKeys&sysparm_query=active=true^category=hardware
```

**get**

Get a record directly by specifying the sys_id in a sysparm_sys_id parameter.

```plaintext
https://<instance name>.service-now.com/incident.do?
JSONv2&sysparm_sys_id=9d385017c611228701d22104cc95c371
```
Optionally, you may also specify the `sysparm_action` parameter:

```
https://<instance name>.service-now.com/incident.do?
JSONv2&sysparm_action=get&sysparm_sys_id=9d385017c611228701d22104cc95c371
```

**getRecords**

Get all records by specifying an encoded query string in the `sysparm_query` parameter.

```
https://<instance name>.service-now.com/incident.do?
JSONv2&sysparm_action=getRecords&sysparm_query=active=true^category=hardware
```

**Data modification API**

Modify data using the JSON web service by sending an HTTPS POST request to the instance.

The HTTP POST must contain a `sysparm_action` parameter to indicate the type of action to be performed, with the incoming JSON object post in the body.

**Note:** The content-type of the POST should be application/json. It cannot be application/x-www-form-urlencoded or multipart/form-data.

**insert**

Create a new record in ServiceNow. The JSON object has to be POSTed as the body (content-type is usually application/json, although not enforced). The response from the record creation is a JSON object of the incident that was created.

For example, posting the following JSON object:

```
{ "short_description": "this is a test", "priority": "1" }
```

to the following URL:

```
https://your_instance.service-now.com/incident.do?
JSONv2&sysparm_action=insert
```

creates an incident.

Optionally, you may also specify the `sysparm_action` in the JSON object. The parameter inside the JSON object takes precedence over the URL parameter. For example:

```
{ "sysparm_action": "insert", "short_description": "this is a test", "priority": "1" }
```

**insertMultiple**

To create multiple new records in ServiceNow, the input JSON object for the insert function must be an array. The response from the record creation is a JSON object of the incidents that were created. For example, the following JSON object:

```
{ "records" : [ { "short_description" : "this was inserted with python using JSON 1" , "priority" : "1 - Critical" , "impact" : "1" , "caller_id" : "Fred Luddy" } , { "short_description" : "this was inserted with
```
posted to one the following URLs:

https://<instance name>.service-now.com/incident.do?JSONv2&sysparm_action=insert
https://<instance name>.service-now.com/incident.do?JSONv2&sysparm_action=insertMultiple

creates two incidents. Note the fields described as an array value for the records field.

**update**

Update a record or a list of records filtered by an encoded query string specified by the sysparm_query URL parameter. The JSON object has to be posted as the body (content-type is usually application/json, although not enforced). The response from the record creation is an array of JSON objects representing the records that were updated.

For example, posting the following JSON object:

```json
{"short_description":"this was updated with python", "priority": "3", "impact": "1"}
```

to the following URL:

https://instance_name.service-now.com/incident.do?JSONv2&sysparm_query=priority=3&sysparm_action=update

updates all incidents with priority 3, and sets the values specified by the JSON object.

**deleteRecord**

Delete a single record from the targeted table, identified by a sysparm_sys_id parameter. The parameter may be encoded in the input JSON object or given as a URL parameter.

For example, posting:

```json
{"sysparm_sys_id":"fd4001f80a0a0b380032ffa2b749927b"}
```

to the following URL:

http://instance_name.service-now.com/incident.do?JSONv2&sysparm_action=deleteRecord

deletes the incident record identified by the sys_id fd4001f80a0a0b380032ffa2b749927b.

**deleteMultiple**

Delete multiple records from the targeted table, filtered by an encoded query string specified in the sysparm_query URL parameter. The filter may also be encoded in the input JSON object.

For example, posting:

```json
{"sysparm_query":"short_description=this was updated with python"}
```
to the following URL:

```
http://instance_name.service-now.com/incident.do?JSONv2&sysparm_action=deleteMultiple
```
deletes all incident records where the `short_description` field contains the value "this was updated with python".

**PDF web service**

The ServiceNow platform supports programmatic retrieval of PDF data through an HTTP GET request. The request is triggered when you use the `PDF` URL parameter.

For example, to retrieve a list of "incidents" and save to a PDF file, issue an HTTP GET to the URL:

```
https://instance_name.service-now.com/incident_list.do?PDF
```

The "content-disposition" HTTP header in the response indicates the file name and extension of the extract. In the example, it will be "incident.pdf"

**Note:** For PDF export, there is a distinction between targeting a table and targeting its list. To generate a PDF of a list of records, suffix the target with "_list". To target a single record, you must specify the `sys_id` parameter to identify the record for which you are generating the PDF.

**PDF web service parameters**

The following URL parameters customize and filter the PDF content that is returned by an HTTP GET request.

### `sysparm_query`

The value of the `sysparm_query` parameter is an encoded query string. Use the parameter to filter the data using the encoded query before returning the PDF content. The following request filters the list to return only active incidents:

```
https://instance_name.service-now.com/incident.do?PDF&sysparm_query=active=true
```

### `sysparm_view`

The value of the `sysparm_view` parameter specifies which list view to use to limit the field values that are returned. For example, to use the "ess" view, use:

```
https://instance_name.service-now.com/incident.do?PDF&sysparm_view=ess
```

**To generate a list,** use:

```
https://instance_name.service-now.com/incident_list.do?PDF&sysparm_view=ess
```

**XML web service**

The ServiceNow platform supports programmatic retrieval of XML data through a URL query (HTTP GET request).

The request is triggered by use of a URL parameter. The following XML-based export formats from a URL query are supported.
Table 1090: XML-based export formats

<table>
<thead>
<tr>
<th>Export Processor</th>
<th>URL Syntax</th>
<th>Example URL</th>
</tr>
</thead>
</table>

An example XML document retrieved using a URL with the following pattern https://instancename.service-now.com/change_request.do?XML is displayed below. Access control is protected using Basic Authentication.

```xml
<xml>
  <change_request>
    <description>Please install new Cat. 6500 in Data center 01</description>
    <category>Hardware</category>
    <due_date/>
    <scope>3</scope>
    <comments/>
    <knowledge>false</knowledge>
    <active>true</active>
    <phase>requested</phase>
    <justification/>
    <cab_date/>
    <review_date/>
    <sys_created_on>2009-04-14 23:14:14</sys_created_on>
    <correlation_id/>
    <follow_up/>
    <sys_domain>global</sys_domain>
    <close_notes/>
    <urgency>3</urgency>
    <change_plan/>
    <business_duration/>
    <sys_id>46cb2f54a9fe198101cf6814a2754606</sys_id>
    <state>1</state>
    <reason/>
    <closed_at/>
    <closed_by/>
    <sys_updated_on>2009-12-21 23:47:15</sys_updated_on>
    <contact_type>phone</contact_type>
    <group_list/>
    <risk>3</risk>
    <assignment_group/>
    <sys_updated_by>pat.casey</sys_updated_by>
    <production_system>false</production_system>
    <approval_history/>
    <sys_created_by>glide.maint</sys_created_by>
    <end_date>2009-12-21 00:30:00</end_date>
    <order/>
    <variables/>
    <calendar_duration/>
    <work_start/>
    <backout_plan/>
    <start_date>2009-12-20 19:30:00</start_date>
    <correlation_display/>
</change_request>
</xml>

© 2017 ServiceNow. All rights reserved.
XML web service parameters

ServiceNow supports the URL parameters, sysparm_query and useUnloadFormat, to customize and filter the response you get.

Table 1091: XML web service parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysparm_query</td>
<td>The value for this URL parameter is that of an encoded query string and when used, will filter the data using the encoded query before returning the XML file. The following request will filter the list to only return &quot;incidents&quot; that are active:</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>useUnloadFormat</td>
<td>This parameter indicates that the XML format that is returned to be of an unload format. The unload format is the same format you get when you from a list in the UI, you right-click a list header and select <strong>Export &gt; XML &gt; .</strong> The unload-formatted XML can be imported back into the instance by administrators of your system. To enable the unload format from a URL, use the useUnloadFormat=true URL parameter. For example: <a href="https://instance_name.service-now.com/incident.do?XML&amp;useUnloadFormat=true">https://instance_name.service-now.com/incident.do?XML&amp;useUnloadFormat=true</a></td>
</tr>
</tbody>
</table>

Using tyhe URL parameter produces an XML format that looks like:

```xml
<unload unload_date="2010-03-12 17:48:30">
  <incident action="INSERT_OR_UPDATE">
    <description/>
    <category>inquiry</category>
    <due_date/>
    <comments/>
    <knowledge>false</knowledge>
    <business_stc>15261</business_stc>
    <active>false</active>
    <sys_created_on>2010-01-02 22:51:33</sys_created_on>
    <correlation_id/>
    <follow_up/>
    <sys_domain>global</sys_domain>
    <close_notes/>
    <urgency>1</urgency>
    <incident_state>7</incident_state>
    <business_duration>1970-01-01 04:14:21</business_duration>
    <sys_id>46b66a40a9fe198101f243dfbc79033d</sys_id>
    <state>1</state>
    <closed_at>2009-12-28 22:55:16</closed_at>
    <closed_by display_value="Don Goodliffe">9ee1b13dc6112271007f9d0efdb69cd0</closed_by>
    <sys_updated_on>2010-04-12 01:20:57</sys_updated_on>
    <contact_type>phone</contact_type>
    <group_list/>
    <assignment_group display_value=""/>
    <sys_updated_by>glide.maint</sys_updated_by>
    <problem_id display_value="PRB12345">9ee1b13dc6112271007f9d0efd2211d0</problem_id>
    <approval_history/>
    <sys_created_by>don.goodliffe</sys_created_by>
    <order/>
    <caller_id display_value="Rick Berzle">5137153cc611227c000bbd1bd8cd2006</caller_id>
    <variables/>
    <calendar_duration>1970-01-02 00:04:53</calendar_duration>
    ...
  </incident>
</unload>
```
RSS web service

RSS (Rich Site Summary) is a format for delivering web-based information that changes regularly.

RSS feed generator

ServiceNow supports the dynamic generation of RSS feeds.

Much like our Web Services implementation, the retrieval of an RSS feed representation of information is simply done by specifying an RSS parameter at the end of the URL to a table list. For example, the following will return a list of all incidents in RSS 2.0 format:

Adding a Query

To associate a query to the list so that a filtered list is returned, use the sysparm_query parameter. For example, the following will return a list of all incidents where the priority field is 1 (Critical):

```url
https://<instance name>.service-now.com/incident.do?
sysparm_query=priority=1&RSS
```

If you have a multi part query then you would separate the parts with the ^ character. For example to get all priority 1 incidents with a category of software you would:

```url
https://<instance name>.service-now.com/incident.do?
sysparm_query=priority=1^category=software&RSS
```

If you want to query on a field that is a reference to another file then you need to use javascript to resolve the reference to the other file. For example, the assigned_to field in incident is a reference to a user record. If you wanted to find all the incidents assigned to "ITIL User" then you would do the following:

```url
https://<instance name>.service-now.com/incident.do?
sysparm_query=assigned_to=javascript:GetIDValue('sys_user','ITIL%20User')&RSS
```

Note: You can in most cases simply append "&RSS" to a URL that you generate in the UI or that of your favorite module. The easiest way to get the URL is to simply click the last breadcrumb from the list view. After appending "&RSS" then you can use this URL in your RSS feed reader.
Limiting results with a view

The description element in the returned RSS xml is constructed using the view as specified in the URL, when no view is specified, the default no-name view is used.

To change this format, specify the sysparm_view parameter on the URL. For example, the following request will return the incidents list. However the result will be restricted to only the fields available in the ess view:

https://<instance name>.service-now.com/incident.do?sysparm_query=priority=1&sysparm_view=ess&RSS

Additionally, the RSS item title can be modified using the sysparm_title_view URL parameter. When specified, the item title will be constructed using the fields specified in the view. For example:


Formatting results

The description element in the returned RSS xml can be formatted by setting the URL parameter sysparm_format=true and specifying the format string in the property glide.rss.description_format.

By default, when the URL parameter is present, the description element will be formatted to contain the field label and value using the following format string:

<b>{1}</b>: {2}<br/>

- {0} - field name
- {1} - field label
- {2} - field value

This default format string can be overridden using the property glide.rss.description_format. An example of the formatted RSS feed can be seen in the following screen capture from Firefox:
To enforce basic authentication on each request for an RSS feed, set the property glide.basicauth.required.rss to true. RSS request would have to contain the Authorization header as specified in the Authentication protocol. Because the request is non-interactive, we always require the Authorization header during a request.

**Note:** If you plan to disable RSS basic authentication, make sure that tables in the platform have the right ACL entries to protect from unauthorized access.

To specify basic authentication on the URL, put the username and password pair separated by a colon in front of the server name before an @ character. For example, to submit the demo credentials for the ITIL user, use the following URL.

```
https://itil:itil@<instance name>.service-now.com/incident.do?RSS
```

Some older browsers, such as Microsoft IE 7 do not support direct URL authentication. If the site uses basic authentication, Internet Explorer automatically prompts users for a user name and a password. In some cases, users can click the Remember my password box in the prompt to save their credentials for later visits to that site.

**RSS title override**

You may optionally override the automatically generated title of the RSS feed by added the sysparm_title parameter to the request URL.
For example, you can specify the title *Priority One Incidents* using the following request URL.

```plaintext
https://<instance name>.service-now.com/incident.do?
sysparm_query=priority=1&sysparm_view=ess&RSS=sysparm_title=Priority%20One%20Incidents
```

This will produce results as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>Priority One Incidents</title>
    <link>http://www.service-now.com/demo/nav_to.do?url=incident.do?sysparm_query=priority=1</link>
    <description>Priority One Incidents</description>
    <copyright>Copyright 2006 Service-now.com</copyright>
    <pubDate>Mon, 12 Jun 2006 11:04:44 PDT</pubDate>
    <lastBuildDate>Mon, 12 Jun 2006 11:04:44 PDT</lastBuildDate>
    <generator>jRSSGenerator by Henrique A. Vieceli</generator>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <item>
      <title>INC000009</title>
      <link>http://www.service-now.com/demo/nav_to.do?url=incident.do?sysparm_query=priority=1&sysparm_view=incident_list.do%3Bsysparm_query=active=true</link>
      <description>INC000009 2006-02-01 14:50:23 Reset my password</description>
      <author>glide.maint</author>
      <guid>46b66a40a9fe198101f243dfbc79033d</guid>
      <pubDate>Wed, 17 May 2006 18:20:57 PDT</pubDate>
    </item>
    <item>
      <title></title>
      <link></link>
      <description></description>
      <author></author>
      <guid></guid>
      <pubDate></pubDate>
    </item>
  </channel>
</rss>
```

**Figure 835: RSS Out**

**RSS feed reader**

Create a scrolling RSS feed reader using a UI page.

You must create a feed parser using an RSS API, such as the Google Feed API.

Developer's guide: http://code.google.com/apis/ajaxfeeds/documentation/


**Scrollable areas**

A scrollable area is a div where contents scroll from the bottom up over time.

You can scroll any HTML content, and anything inside the scroller is operational HTML with functioning links and images.
To make a scrollable areas, wrap the scrolling content inside of a scrollable_area tag, likely in a UI Page:

```xml
<g:scrollable_area height="100px">
  <g2:evaluate var="jvar_temp" expression="var kb = new GlideRecord('kb_knowledge');="/>
  <g:inline template="kb_section.xml"/>
</g:scrollable_area>
```

The system will then create a 100 pixel high div and the contents will automatically scroll from bottom to top. If you have a 1000 pixel high block of text, for example, you'll see the top 100 pixels and then pixels 2-101, then 3-102, etc. Once it reaches the top it'll wrap back around to the bottom.

This sample code will create a scroller with a list of priority 1 incidents.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:glide="null">
  <g2:evaluate var="jvar_inc">
    var inc = new GlideRecord('incident');
    inc.addActiveQuery();
    inc.addQuery('priority',1);
    inc.query();
  </g2:evaluate>

  <g2:scrollable_area height="100px">
    <table border="0" cellspacing="2" cellpadding="0" width="100">
      <j2:while test="$[inc.next()]">
        <j2:set var="jvar_inc_link" value="incident.do?sys_id=$[inc.sys_id]"/>
        <j2:set var="jvar_inc_list_link" value="incident_list.do?sysparm_query=active=true"/>
        <tr>
          <td>
            <a href="$[jvar_inc_link]">
              <IMG SRC="images/services.png" style="padding-right:10px"></IMG>
            </a>
            <a href="$[jvar_inc_link]" style="padding-right:10px;color:blue">$[inc.number]</a>
          </td>
          <td>$[inc.short_description]</td>
        </tr>
      </j2:while>
    </table>
    <tr>
      <td align="center" colspan="2"><a href="$[jvar_inc_list_link]" style="color:blue">View all active incidents</a></td>
    </tr>
  </g2:scrollable_area>
</j:jelly>
```

Add scrolling elements in forms
You can add scrolling areas to forms as well as UI pages.

1. Create a UI Macro with the script.
2. Create a Formatter to reference the script.

Priority 1 incidents example
This example scrolling element demonstrates how to create a UI macro to a scrolling list of priority 1 incidents.
Use the following example code:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null"
 xmlns:g2="null">
  <g2:evaluate var="jvar_inc">
    var inc = new GlideRecord('incident');
    inc.addActiveQuery();
    inc.addQuery('priority',1);
    inc.query();
  </g2:evaluate>

  <div style="background-color:DDDDDD; padding-left:10px; line-height:19px;
    border:2px white solid" width="100%" nowrap="true">
    Priority 1 Incidents:
    <input id="make_spacing_ok" style="visibility:hidden;width:0px;"
      title=""/>
  </div>
  <g2:scrollable_area height="100px" width="100%">
    <j2:while test="${inc.next()}">
      <j2:set var="jvar_inc_link" value="incident.do?sys_id=${inc.sys_id}"/>
      <j2:set var="jvar_inc_list_link" value="incident_list.do?sysparm_query=active=true"/>
      <span style="line-height: 10px; padding-left:10px">
        <a href="${jvar_inc_link}"
          style="padding-right:10px; color:blue">${inc.number}</a>
        ${inc.short_description}<br style="line-height:5px"/>
      </span>
    </j2:while>
  </g2:scrollable_area>
</j:jelly>
```

Navigate to **System UI > Formatters** and create a Formatter that refers to the UI Macro above.

**Add a scrolling news panel to your homepage**

To add a scrolling news panel to your homepage:

1. Navigate to a homepage.
2. Click **Add Content** to get a list of possible content types.
3. Click **Scrollers** in the leftmost box of the add content dialog.
4. Click the news scroller.
5. Click **Add** to add it on your page.
News in the scroller
The news scroller gets its news list by going to the Knowledge Base and querying the short descriptions of any items there with a topic of "News".

To add news, go to the Knowledge Base and add new entries with a topic of "News".

Style Control

If you want to make a headline standout in the news scroller, use HTML tags on the short description. For example, enter `<b>My Bold Test</b>` and My Bold Test will appear in the scroller.

Note: The property glide.ui.escape_text must be false to use HTML formatting.

Scrolling News Panel Notes and Limitations

Scrollable news panels offer a way for you to display high impact news items on your users home pages in an eye catching scrolling news display.

- Scrolling is currently always from the bottom to the top.
- Scrolling is 1 pixel a at a time every 100 ms.
- You can have as many scrollable areas on a screen as you want (they get system generated unique names).
- Works in Internet Explorer and Firefox.
- If the object you are scrolling is shorter than the scrollable area, it still scrolls.
- If you mouse over a scrollable area, it stops scrolling (so you can click something without chasing it as it scrolls).

RSS feed reader example
An example of how to set up an RSS feed reader using a CNN RSS feed.

Step 1. Create UI Page

Set up a UI page named 'render_gadget_widgetRSS' with the following HTML

```xml
<?xml version= "1.0" encoding= "utf-8" ?>
<j:jelly trim = "false" xmlns:j = "jelly:core" xmlns:g = "glide"xmlns:j2 = "null" xmlns:g2 = "null" ><style
   type = "text/css" >
   #feedControl .gf-title {
     color:#0000ff;
   }
 </style>
<g2:scrollable_area height = "100px" ><div id = "feedControl"
  >Loading...</div><script type = "text/javascript" >
google.load("feeds", "1");

function initialize() {
```

© 2017 ServiceNow. All rights reserved. 3401
var feedControl = new google.feeds.FeedControl();
feedControl.addFeed("http://rss.cnn.com/rss/cnn_topstories.rss", "CNN");
feedControl.setLinkTarget(google.feeds.LINK_TARGET_BLANK);
feedControl.setNumEntries(5);
feedControl.draw(document.getElementById("feedControl"));
}

if(typeof google == "undefined" || navigator.appName == "Microsoft Internet Explorer"){
    //Give us some time to load the script
    setTimeout('initialize()',1500);
}
else{
    initialize();
}
</script></g2:scrollable_area></j:jelly>

### Step 2. Create UI Widget

Create a new javascript widget under 'System UI > Widgets' with the following script. You can replace the 'RSS Feed Sample' line with the title of your widget. 'widgetRSS' should be replaced with whatever comes after 'render_gadget_' in the name of your UI page.

```javascript
function sections () { return { 'RSS Feed Sample' : { 'type' : 'widgetRSS' } } };

function render () { var type = renderer.getPreferences().get( "type" );
    var gf = new GlideForm(renderer.getGC(), "render_gadget_" + type, 0);
    gf.setDirect( true );
    gf.setRenderProperties(renderer.getRenderProperties()); return gf.getRenderedPage(); }

function getEditLink () { return "sys_ui_page.do?
    sysparm_query=name=render_gadget_" + renderer.getPreferences().get( "type" ); }
```

### Step 3. Create Global UI Script

Create a UI script under System UI > UI Scripts with the following script. Select the Global checkbox on the UI Script form. The script must include your Google API key.

```javascript
document.write('<script type="text/javascript" src="https://www.google.com/jsapi?key=your_instance_api_key_here"></script>');
document.write('<script type = "text/javascript" >google. load ( "feeds" , "1" );</script >');
```

You can now navigate to your homepage and add the widget. Below is a sample showing what the widget looks like.
Web services security

Web services security is enforced using a combination of basic authentication challenge/response for the HTTP protocol and system-level access control using the Contextual Security Manager.

To enforce basic authentication on each Web Service request, each request must contain the **Authorization** header as specified in the **Basic Authentication** protocol. Because the request is non-interactive, the **Authorization** header is required in a request.

There is an added advantage when you supply basic authentication information whether or not it is required: the data that is created or updated as a result of the Web service invocation is done on behalf of the user supplied in the basic authentication credentials. For example, when creating an Incident record, the **journal** fields will contain the user ID of basic authenticated user, instead of the default "Guest" user.

Outbound web services

Outbound web services allow you to access remote endpoints and perform web service requests from a ServiceNow instance.

Outbound REST web service

ServiceNow outbound REST functionality allows you to retrieve, create, update, or delete data on a web services server that supports the REST architecture.

A REST message can be sent by a REST workflow activity or by using the RESTMessageV2 script API. You can run REST messages from a MID Server which allows the message to communicate with REST providers on an internal network.

ServiceNow REST functionality is flexible enough to accommodate many web service APIs. Be sure you are familiar with your web service and the parameters it accepts before attempting to define a REST message in ServiceNow.

The following video tutorial demonstrates how to configure outbound REST web service messages to consume third-party web services from the ServiceNow platform.

REST message elements

An outbound REST message is composed of several elements, such as the endpoint and HTTP methods. A REST message contains the following elements:
Create a REST message

You can send requests to a REST web service endpoint by creating a REST message record.

Role required: web_service_admin

1. Navigate to System Web Services > REST Message.
2. Click New.
3. Complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a descriptive name for this message.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>Enter the endpoint that this REST message is sent to. The endpoint value may include variables using the format <code>${variable}</code>.</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
Authentication type | Select the type of authentication to use, if any, and the profile record that contains the user credentials.
Outbound REST supports basic authentication and OAuth 2.0. Outbound REST supports mutual authentication with basic authentication only.
Authentication configured here is inherited by the associated HTTP methods. You can configure authentication for each method which overrides any authentication setting at the message level.

HTTP Headers | Double-click a row in the HTTP Headers embedded list to define the header Name and Value.
The web service provider determines which headers are supported or required.
See List of HTTP Header Fields for a list of HTTP header fields.

4. **Click Submit.**

   ServiceNow saves the record and automatically creates HTTP methods (get, put, post, and delete).

After creating the REST message, you can define new HTTP methods and run a request.

*Define a REST message HTTP method*

Define an HTTP method such as GET or POST to send a request to a web service provider.

Role required: web_service_admin

When you create a REST message record, several default HTTP methods are automatically created using settings inherited from the REST message record, such as the **Endpoint**. You can create additional HTTP methods or modify the default HTTP methods to implement new behavior.

1. **Navigate to System Web Services > Outbound > REST Message.**
2. **Select a REST message you want to define an HTTP method for.**
3. **In the HTTP Methods related list, click New.**
4. **Enter the Method you want to implement, such as GET or POST.**
5. **Enter the Endpoint this HTTP method should access.**
   - The endpoint value may include variables using the format ${variable}.
6. **Right-click the form header and select Save.**

   After creating the HTTP method, you can override the security settings from the parent REST message, configure HTTP headers, add variables, or test the method. For PUT, POST, and PATCH methods you can define a message body.

*Testing REST message HTTP methods*

After configuring an HTTP method for an outbound REST message, you can test it to ensure that the request is valid and the response returns as expected.

To test an HTTP method, click the **Test** related link on the HTTP Method form.
Each test run displays the response status, such as 200 for a successful GET request, the full endpoint URL, any parameters passed in the request, and the response body.

**Note:** Fields on the Test Runs form are for information only; changes to these fields do not apply to the REST message or HTTP method. Do not modify these values when testing different REST message configurations. Instead, update the REST message or HTTP method, then run a new test.

If the HTTP method includes variables, the **Test value** for each variable in the **Variable Substitutions** related list is used when testing the method.

Completed test runs for an HTTP method appear in the **Test Runs** related list.

**Define a REST message HTTP header**

Define an HTTP header for a REST message or HTTP method to send that header with REST requests.

Role required: web_service_admin

You can specify an HTTP header for a REST message, or for an HTTP method. Headers defined for a REST message apply to all HTTP methods for that REST message. If you specify the same header for both a REST message and a child HTTP method, the value defined for the HTTP method overrides the value from the parent REST message.

1. Navigate to **System Web Services > REST Message**.
2. Select a REST message.
3. Optional: To specify a header for an HTTP method instead of the REST message, in the **HTTP Method** related list, select an HTTP method.
4. Select the **HTTP Request** tab.
5. In the **HTTP Headers** embedded list, click **Insert a new row**.
6. Enter the name of the header, such as Content-Type or Accept.
   
   Supported headers depend on the REST web service provider you want to connect to. Refer to the documentation for your web service provider to identify which headers are valid or required.

   7. Click on the **Value** field for the new row and enter the value you want to assign this header.

   You can use a variable in the format ${variable} instead of a static value. You can assign a value to the variable when sending a REST request.

8. Click **Update**.

**Sending outbound REST messages through a MID Server**

You can configure a REST message HTTP method to be sent through a MID Server.

By using a MID Server, the request can reach an endpoint that is behind a firewall or within a private network.

To configure an HTTP method to use a MID Server, select a MID Server in the **Use MID Server** field on the HTTP Method form. The instance must have an active MID Server to use this functionality.

**Using special characters in URLs**

A REST function URI or function variable may use special characters, such as pipe (|) characters.

When using these characters in a REST message, use URL encoding to escape these characters. For example, to use a parameter value of user|title, enter user%7Ctitle. Entering special characters directly may cause the REST message to fail and display the response Invalid uri <URI>: Invalid query.

**Outbound REST authentication**

Outbound REST messages support multiple types of authentication.

Different web service providers may require a specific type of authentication.

Outbound REST supports the following authentication formats.
Authentication requirements

Outbound REST supports mutual authentication only when using basic authentication. Mutual authentication is not available with OAuth 2.0.

OAuth 2.0 can be used only with messages that are not configured to use a MID Server. You cannot send OAuth 2.0 authenticated messages through a MID Server.

When scripting new REST messages configured with authentication you must use the RESTMessageV2 API. The legacy RESTMessage APIs do not support current authentication formats.

Configure a REST message with basic auth
You can configure an outbound REST message to provide basic authentication credentials with each request.

Role required: web_service_admin

Before starting this procedure, ensure there is a REST Message record that you want to configure to use basic auth.

Note: Ensure any scripts that call this REST message use the RESTMessageV2 API. The RESTMessageV2 API is required to send authenticated REST messages via scripts.

1. Navigate to System Web Services > Outbound > REST Message.
2. Select a REST message record.
3. In the Authentication type field, select Basic.

Note: The Basic (Simple) choice appears on REST message records configured to use basic authentication prior to the Geneva release. This choice is intended for compatibility with older REST messages and should not be used for REST messages created in the Geneva release or later.

4. In the Basic auth profile field, select the basic authentication profile that contains the credentials you want to send.
5. Click Submit.

Test the REST message to ensure you receive the expected response. You can optionally specify different authentication settings for each HTTP method related to this REST message, overriding the parent REST message settings.

Create a basic auth profile
Create a basic auth profiles to specify basic authentication credentials for one ore more REST messages.

Role required: web_service_admin

1. Navigate to System Web Services > REST Message.
2. Select a REST message record.
3. In the Authentication type field, select Basic.
4. In the Basic auth profile field, click the reference lookup icon.
5. Click New.
6. Enter a descriptive Name for the profile.
7. Enter the Username and Password you want to send as basic authentication credentials.
8. Click Submit.

Configure a REST message to use this basic auth profile.

**Configure a REST message with OAuth**

You can configure an outbound REST message to send OAuth credentials with the request.

Role required: web_service_admin and oauth_admin

Before starting this procedure, ensure:

- There is a REST Message record that you want to configure to use OAuth.
- There is an OAuth provider set up in the OAuth application registry with the OAuth client information to use.
- The OAuth provider has an associated OAuth 2.0 profile.
- The REST message HTTP Methods are not configured to use a MID Server.

**Note:** Ensure any scripts that call this REST message use the RESTMessageV2 API. The RESTMessageV2 API is required to send authenticated REST messages via scripts.

1. Navigate to **System Web Services > Outbound > REST Message**.
2. Select a REST message record.
3. In the **Authentication type** field, select **OAuth 2.0**.
4. In the **OAuth profile** field, select the OAuth 2.0 profile that specifies the credentials you want to send.
5. Right-click the form header and select **Save**.
   An info message appears at the top of the form indicating that you must request a new OAuth token.
6. Click the **Get OAuth Token** related link.
   Depending on your OAuth provider, a separate window from your provider may appear asking for confirmation before providing a token. Complete any steps required by the provider to obtain the token.

Test the REST message to ensure you receive the expected response. You can optionally specify different authentication settings for each HTTP method related to this REST message, overriding the parent REST message settings.

### Outbound REST with OAuth 2.0 profile tutorial - integrating with Google Contacts API

This tutorial explains how to use an OAuth 2.0 profile to authenticate an outbound REST message with Google to retrieve contact information.

The procedures detailed in this tutorial require the oauth_admin and web_service_admin roles. Ensure you have both of these roles before starting this tutorial.

### OAuth 2.0 tutorial - configure the Google service as an OAuth provider

Use the Google Developer Console to set up an OAuth 2.0 provider.

Role required: None

This procedure is performed within the Google Developer Console. You must have a Google account to access this console.

Configure the Google service in order to obtain a client ID and client secret, and specify your ServiceNow instance URL as the OAuth redirect URL.

**Note:** This information describes the state of the Google Developer Console and Contacts API as of July 22, 2015. Changes made after that date may not be included in this document.

1. Navigate to the Google Developer Console ([https://console.developers.google.com](https://console.developers.google.com)).
2. Log in using your Google credentials.
3. Click **Select a project**.
4. Click **Create a project**.
5. Enter a **Project name**.
6. Click **Create**.
   After Google creates the project, the project dashboard appears.
7. Navigate to **APIs & auth > APIs**.
8. Select the **Contacts API**.
9. Click **Enable API**.
10. Navigate to **APIs & auth > Credentials**.
11. Click **Create new Client ID**.
12. Ensure the **Web application** radio button is selected and click **Configure consent screen**.
13. Enter a descriptive **Product name**.
   This name appears when you authorize the OAuth token in your instance.
14. Click **Save**.
15. In the **Create Client ID** window, add the OAuth redirect URI for your instance to the **Authorized redirect URIs** field.
   This URI follows the format `https://<instance>.service-now.com/oauth_redirect.do`
16. Click **Create Client ID**.
   The client ID information appears.
17. Record the **Client ID** and **Client secret** values.
   You will need these values to configure the Google service as an OAuth provider in your instance.

OAuth 2.0 tutorial - create an OAuth provider and profile
Set up the Google service as an OAuth provider in ServiceNow by entering your client information, Google API URLs, and configuring the OAuth profile.

Role required: oauth_admin
You must have configured the Google service as an OAuth provider and recorded your **Client ID** and **Client Secret** values.

1. Navigate to **System OAuth > Application Registry**.
2. Click **New**.
3. Select **Connect to a third party OAuth Provider**.
4. Enter a **Name** for the OAuth provider. For this example, use **Google**.
5. Enter the **Client ID** and **Client Secret** that you obtained from Google.
6. Set the **Default Grant type** to **Authorization Code**.
   This URL must match the redirect URL provided to Google.
10. In the **Token Revocation URL** field, enter `https://accounts.google.com/o/oauth2/revoke`.
11. Right-click the form header and select **Save**.
   A new OAuth Entity Profile record is created.
12. In the **OAuth Entity Scopes** embedded list, add a new row with the **Name** and **OAuth scope** values set to `https://www.googleapis.com/auth/contacts.readonly`.
13. Right-click the form header and select **Save**.
14. In the **OAuth Entity Profiles** embedded list, select the automatically-created profile.
15. In the OAuth Entity Profile Scopes embedded list, add a new row and select the Google contacts API read-only scope.

16. Click Update.

OAuth 2.0 tutorial - create a REST message
Create a REST message and associated HTTP method to contact the Google service using the OAuth 2.0 profile.

Role required: web_service_admin and oauth_admin
You must have configured an OAuth provider and profile using the Google API information and your OAuth credentials.

1. Navigate to System Web Services > REST Message.
2. Click New.
3. Enter a descriptive Name.
4. In the Endpoint field, enter https://www.google.com/m8/feeds/contacts/default/full.
   By using default instead of a specific username, the Google API uses the OAuth credentials to determine which account to get information from.
5. In the Authentication tab, set the Authentication type to OAuth 2.0.
6. In the OAuth profile field, select the Google contacts OAuth profile.
7. Right-click the form header and select Save.
8. Click the Get OAuth Token related link to request an authorization token from Google using the configured client ID and secret.
9. In the Request for Permission window that appears, click Accept to grant access to your Google contacts.
   The token acquired is not directly accessible in your instance.
10. In the HTTP Methods related list, select the GET method.
11. Leave the HTTP method Authentication type as -- None -- to use the OAuth profile from the parent REST message record.
12. On the HTTP Request tab, add a new row to the HTTP Headers related list with a Name of GData-Version and a Value of 3.0.
13. Right-click the form header and select Save.
14. Click the Test related link.
   The test result should display an HTTP Status of 200, and the result of the contacts API call.

Outbound REST mutual authentication
Mutual authentication causes the web service provider and consumer to authenticate with each other before communicating.

ServiceNow supports mutual authentication for outbound web services. Mutual authentication is not available for inbound web services or for outbound web services that use a MID Server.

Variable substitution in outbound REST messages
You can use variables when creating outbound REST messages and assign values to those variables when performing a request.

Variables are allowed in the Endpoint URL, HTTP Header and HTTP Query Parameter Value fields, and the Content field for POST and PUT methods.

The syntax for variables is ${variable_name}. The REST message substitutes this variable with the parameter values provided when the method runs. For example, if the REST message Endpoint is http://myserver.mycompany.com/offices/${id}, a parameter named id must exist and contain a value that can be used when the method runs.
You can assign a value to variables when running the request using the RESTMessageV2 API
`setStringParameter` and `setStringParameterNoEscape` methods.

When testing an HTTP method that includes variables, the **Test value** for each variable in the **Variable Substitutions** related list is used.

**Generate REST message variables**

You can automatically populate the list of variable substitutions, based on variables defined in several REST message HTTP method fields.

Role required: `web_service_admin`

Before starting this procedure, create a REST Message record with at least one HTTP method that uses variables.

1. Navigate to **System Web Services > REST Message**.
2. Select a REST Message record.
3. Select a method from the **HTTP Methods** related list.
4. Click the **Auto-generate variables** related link.
   - The **Variable Substitutions** related list is automatically populated for any variables defined in the HTTP Method **Endpoint** field and the **HTTP Headers** and **HTTP Query Parameters** embedded lists.
   - For POST and PUT messages, variables defined in the **Content** field are also used.

You can use the REST Message workflow activity to send the message, or click **Preview Script Usage** to get a sample script. The sample script includes a `setStringParameter` call for each defined variable substitution that allows you to assign a value to the variable in your script.

**Scripting outbound REST**

You can call REST methods from any place in the platform where scripting is allowed.

For example, you can return data from a REST endpoint using a business rule when an event is triggered. Create a script from scratch or let the REST message preview feature create the script based on content and parameters you provide in the method record.

**Generate a REST message script preview**

You can generate an example script to send a REST message based on content and parameters you provide in the method record.

Role required: `web_service_admin` or `admin`

Generate an example script and use it as a starting point when scripting outbound REST messages.

1. Navigate to **System Web Services > REST Message**.
2. Select a REST message record.
3. In the **HTTP Methods** related list, select an HTTP method record.
4. Ensure the HTTP method is configured as needed, including any variables.
5. Save the record.
6. In the Variable Substitutions related list, assign a value to each variable.
7. Under Related Links, click **Preview script usage**.
   - The instance displays the script that the REST message generated for this method.
8. Copy this script and modify it as needed to use elsewhere in the instance.

Refer to the RESTMessageV2 and RESTResponseV2 APIs for more information on available scripting methods. Outbound REST scripting examples are also available.

**RESTMessageV2**

The RESTMessageV2 API allows you to send outbound REST messages using JavaScript.

Use the RESTResponseV2 API to manage the response returned by the REST provider.

You can use this API in scoped applications, or within the global scope.

**RESTMessageV2 - RESTMessageV2()**

Instantiates an empty RESTMessageV2 object.

When using an object instantiated this way, you must manually specify an HTTP method and endpoint.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
var sm = new sn_ws.RESTMessageV2();
```

**RESTMessageV2 - RESTMessageV2(String name, String methodName)**

Instantiates a RESTMessageV2 object using information from a REST message record.

You must have a REST message record defined before you can use this constructor.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the REST message record.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>methodName</td>
<td>String</td>
<td>The name of the HTTP method to use, such as GET or PUT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The PATCH method is not supported. Use PUT or POST instead.</td>
</tr>
</tbody>
</table>

```
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
```

RESTMessageV2 - execute()
Send the REST message to the endpoint.

**Table 1096: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1097: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTResponse</td>
<td>The response returned by the REST provider.</td>
</tr>
</tbody>
</table>

```
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute(); //Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password.
```

RESTMessageV2 - executeAsync()
Send the REST message to the endpoint asynchronously. The instance does not wait for a response from the web service provider when making asynchronous calls.

**Table 1098: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1099: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTResponse</td>
<td>The response returned by the REST provider.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.executeAsync(); // Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password.
response.waitForResponse(60); // In seconds. Wait at most 60 seconds to get response from ECC Queue/Mid Server // Might throw exception timing out waiting for response in ECC queue.
```

```javascript
RESTMessageV2 - getEndpoint()
Get the URL of the endpoint for the REST message.
```

### Table 1100: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 1101: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The URL of the REST web service provider.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
var endpoint = sm.getEndpoint();
```

```javascript
RESTMessageV2 - getRequestBody()
Get the content of the REST message body.
```

### Table 1102: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1103: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>the REST message body.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var body = sm.getRequestBody();
```

RESTMessageV2 - getRequestHeader(String headerName)
Get the value for an HTTP header specified in the REST message.
By default, this method cannot return the value for a header set automatically by the system. To grant this method access to all headers, set the property glide.http.log_debug to true.

Table 1104: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>headerName</td>
<td>String</td>
<td>The request header you want to get the value for.</td>
</tr>
</tbody>
</table>

Table 1105: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The value of the specified header.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var header = sm.getRequestHeader("Accept");
```

RESTMessageV2 - getRequestHeaders()
Get HTTP headers that were set by the REST client and the associated values.
This method does not return headers set automatically by the system. To configure this method to return all headers, set the property glide.http.log_debug to true.

Table 1106: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1107: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>An Object that maps the name of each header to the associated value.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var headers = sm.getRequestHeaders();
```

RESTMessageV2 - setBasicAuth(String userName, String userPass)
Sets basic authentication headers for the REST message.

Setting security values using this method overrides basic authentication values defined for the REST message record.

Table 1108: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>String</td>
<td>The username you want to use to authenticate the REST message.</td>
</tr>
<tr>
<td>userPass</td>
<td>String</td>
<td>The password for the specified user.</td>
</tr>
</tbody>
</table>

Table 1109: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
sm.setBasicAuth("username","password");
```

RESTMessageV2 - setEccCorrelator(String correlator)
Associate outbound requests and the resulting response record in the ECC queue. This method only applies to REST messages sent through a MID Server.

The correlator provided populates the Agent correlator field on the ECC queue record for the response. Provide a unique correlator for each outbound request to associate the correct results in the ECC queue with the request when designing asynchronous automation through a MID Server.
### Table 1110: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlator</td>
<td>String</td>
<td>A unique identifier</td>
</tr>
</tbody>
</table>

### Table 1111: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance", "get"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setEccCorrelator("unique_identifier");
```

RESTMessageV2 - `setEccCorrelator(String name, String value)`

Override a value from the database by writing to the REST message payload. This method only applies to REST messages sent through a MID Server.

Use this method when a value from the REST message in the database is invalid, such as when the endpoint URL is longer than the maximum REST endpoint field length. You can set only the endpoint URL using this method by passing `source` as the name parameter.

### Table 1112: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the parameter, such as <code>source</code>.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign to the specified parameter.</td>
</tr>
</tbody>
</table>

### Table 1113: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance", "get"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setEccParameter("source", "http://very.long.endpoint.url");
```

RESTMessageV2 - `setEndpoint(String endpoint)`

Set the endpoint for the REST message.
By default, the REST message uses the endpoint specified in the REST message record. Use this method to override this default. You must call this method when using the `RESTMessageV2 - RESTMessageV2()` on page 3412 constructor with no parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endpoint</td>
<td>String</td>
<td>The URL of the REST provider you want to interface with.</td>
</tr>
</tbody>
</table>

Table 1115: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2();
sm.setEndpoint("http://web.service.endpoint");
```

**RESTMessageV2 - setHttpMethod(String method)**

The HTTP method this REST message performs, such as GET or PUT.

You must set an HTTP method when using the `RESTMessageV2 - RESTMessageV2()` on page 3412 constructor with no parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>String</td>
<td>The HTTP method to perform.</td>
</tr>
</tbody>
</table>

**Note:** The PATCH method is not supported. Use PUT or POST instead.

Table 1117: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2();
sm.setHttpMethod("post");
```

**RESTMessageV2 - setHttpTimeout(Number timeoutMs)**
Set the amount of time the REST message waits for a response from the web service provider before the request times out.

**Table 1118: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeoutMs</td>
<td>Number</td>
<td>The amount of time, in milliseconds, before the call to the REST provider times out.</td>
</tr>
</tbody>
</table>

**Table 1119: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setHttpTimeout(6000);
```

**RESTMessageV2 - setMIDServer(String midServer)**
Configure the REST message to communicate through a MID Server.

**Table 1120: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>midServer</td>
<td>String</td>
<td>The name of the MID Server to use. Your instance must have an active MID Server with the specified name.</td>
</tr>
</tbody>
</table>

**Table 1121: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setMIDServer("mid_server_name");
```

**RESTMessageV2 - setMutualAuth(String profileName)**
Set the mutual authentication protocol profile for the REST message.
Setting a protocol profile using this method overrides the protocol profile selected for the REST message record.

### Table 1122: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>profileName</td>
<td>String</td>
<td>The Name of the protocol profile to use for mutual authentication.</td>
</tr>
</tbody>
</table>

### Table 1123: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
sm.setMutualAuth("mutual_auth_profile_name");
```

**RESTMessageV2 - setQueryParameter(String name, String value)**

Append a parameter to the end of the request URL with the form name=value.

For example, the code

```java
setQueryParameter(“sysparm_query”,“active=true^ORDERBYnumber^ORDERBYDESCcategory”);
```

appends the text sysparm_query=active=true^ORDERBYnumber^ORDERBYDESCcategory to the request URL.

### Table 1124: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the URL parameter to pass.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign the URL parameter.</td>
</tr>
</tbody>
</table>

### Table 1125: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.RESTMessageV2();
//Set up message, including endpoint and authentication
sm.setQueryParameter("sysparm_query","active=true^ORDERBYnumber^ORDERBYDESCcategory");
```
RESTMessageV2 - setRequestBody(String body)
Set the body content to send to the web service provider when using PUT or POST HTTP methods.

When you set the body content using this method, variables in the body are not substituted for parameters from the REST message function record. You must explicitly define all values within the REST message body.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>String</td>
<td>The request body to send.</td>
</tr>
</tbody>
</table>

Table 1127: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Update user","post"); //Might throw exception if message doesn't exist or not visible due to scope.
var body = "<Message body content>";
sm.setRequestBody(body);
```

RESTMessageV2 - setRequestHeader(String name, String value)
Set an HTTP header in the REST message to the specified value.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the header.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign to the specified header.</td>
</tr>
</tbody>
</table>

Table 1129: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
sm.setRequestHeader("Accept","Application/json");
```

RESTMessageV2 - setStringParameter(String name, String value)
Set a REST message function variable with the specified name from the REST message record to the specified value.

XML reserved characters in the value are converted to the equivalent escaped characters. Use `setStringParameterNoEscape` to set a variable without escaping XML reserved characters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the REST message variable. This parameter must be defined in the REST message record before you can assign a value to it.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign the variable.</td>
</tr>
</tbody>
</table>

**Table 1131: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
sm.setStringParameter("s","NOW");
```

**RESTMessageV2 - setStringParameterNoEscape(String name, String value)**

Set a REST message function variable with the specified name from the REST message record to the specified value.

This method is equivalent to `setStringParameter` but does not escape XML reserved characters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the REST message variable. This parameter must be defined in the REST message record before you can assign a value to it.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign the variable.</td>
</tr>
</tbody>
</table>
Table 1133: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setStringParameterNoEscape("s","NOW");
```

RESTMessageV2 - saveResponseBodyAsAttachment(String tableName, String recordSysId, String fileName)

Configure the REST message to save the returned response body as an attachment record.

When you use this function with a REST message that is sent through a MID server, the MID server user must have any roles required to read and write attachment records, as well as any roles required to read and write records on the table specified in the tableName parameter.

The response body does not need to be a binary file to be saved as an attachment. Response bodies using text formats, such as JSON or XML can also be saved. If the instance fails to save the attachment, call `getErrorMessage()` on the related RESTResponseV2 object for error details.

### Table 1134: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>Specify the table that contains the record you want to attach the saved file to.</td>
</tr>
<tr>
<td>recordSysId</td>
<td>String</td>
<td>Specify the sys_id of the record you want to attach the saved file to.</td>
</tr>
<tr>
<td>fileName</td>
<td>String</td>
<td>Specify the file name to give to the saved file.</td>
</tr>
</tbody>
</table>

### Table 1135: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```
(function sampleRESTMessageV2() {
  try{
    var request = new sn_ws.RESTMessageV2();
    request.setHttpMethod('get');

    var attachment_sys_id = '<attachment_record_sys_id>',
                            tablename = 'incident',
                            recordSysId = '<incident_sys_id>',
                            response,
                            httpResponseStatus,
```
```javascript
filename = '<filename>'; //endpoint - ServiceNow REST Attachment API
request.setEndpoint('https://<instance_name>.service-now.com/api/now/attachment/' + attachment_sys_id + '/file');
request.setBasicAuth('<username>', '<password>');

//RESTMessageV2 - saveResponseBodyAsAttachment(String tableName, String recordSysId, String fileName)
request.saveResponseBodyAsAttachment(tablename, recordSysId, filename);

response = request.execute();
httpResponseStatus = response.getStatusCode();

gs.print(" http response status code: " + httpResponseStatus);
} catch(ex){
    var message = ex.getMessage();
gs.print(message);
}
})();
```

**RESTMessageV2 - saveResponseBodyAsAttachment(String tableName, String recordSysId, String fileName, String encryptContext)**

Configure the REST message to save the returned response body as an encrypted attachment record.

When you use this function with a REST message that is sent through a MID server, the MID server user must have any roles required to read and write attachment records, as well as any roles required to read and write records on the table specified in the tableName parameter.

The response body does not need to be a binary file to be saved as an attachment. Response bodies using text formats, such as JSON or XML can also be saved. If the instance fails to save the attachment, call getErrorMessage() on the related RESTResponseV2 object for error details.

**Table 1136: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableName</td>
<td>String</td>
<td>Specify the table that contains the record you want to attach the saved file to.</td>
</tr>
<tr>
<td>recordSysId</td>
<td>String</td>
<td>Specify the sys_id of the record you want to attach the saved file to.</td>
</tr>
<tr>
<td>fileName</td>
<td>String</td>
<td>Specify the file name to give to the saved file.</td>
</tr>
<tr>
<td>encryptContext</td>
<td>String</td>
<td>Specify the sys_id of an encryption context. The saved file is encrypted using this context.</td>
</tr>
</tbody>
</table>
Table 1137: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

RESTMessageV2 - setRequestBodyFromAttachment(String attachmentSysId)
Set the request body using an existing attachment record.

When you use this function with a REST message that is sent through a MID server, the MID server user must have any roles required to read attachment records.

Table 1138: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachmentSysId</td>
<td>String</td>
<td>The sys_id of the Attachment [sys_attachment] record you want to send in this REST message.</td>
</tr>
</tbody>
</table>

Table 1139: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

(function sampleRESTMessageV2() {
    try {
        var request = new sn_ws.RESTMessageV2();
        request.setHttpMethod('post');
        request.setEndpoint('<web service endpoint URL>');
        request.setRequestBodyFromAttachment('<attachment sys_id>');
        var response = request.execute();
        var httpResponseStatus = response.getStatusCode();
        gs.print("http response status_code: " + httpResponseStatus);
    }
    catch (ex) {
        var message = ex.getMessage();
        gs.print(message);
    }
})();

RESTMessageV2 - setAuthenticationProfile(String type, String profileId)
Set the credentials for the REST message using an existing basic auth or OAuth 2.0 profile.
Table 1140: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>String</td>
<td>The type of authentication profile to use. Valid values are 'basic' to use basic authentication, or 'oauth2' to use OAuth 2.0.</td>
</tr>
<tr>
<td>profileId</td>
<td>String</td>
<td>The sys_id of an authentication profile record. When using basic auth, specify the sys_id of a Basic Auth Configuration [sys_auth_profile_basic] record. When using OAuth 2.0, specify the sys_id of a OAuth Entity Profile [oauth_entity_profile] record.</td>
</tr>
</tbody>
</table>

Table 1141: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var requestBody;
var responseBody;
var status;
var sm;
try{
  // Might throw exception if message doesn't exist or not visible due to scope.
  sm = new sn_ws.RESTMessageV2("Yahoo Finance", "get");

  //set auth profile to an OAuth 2.0 profile record.
  sm.setAuthenticationProfile('oauth2', '1234adsf12321231123qasdfs');

  sm.setStringParameter("symbol", "NOW");
  sm.setStringParameterNoEscape("xml_data", "<data>test</data>");
  //In milliseconds. Wait at most 10 seconds for response from http request.
  sm.setHttpTimeout(10000);

  //Might throw exception if http connection timed out or some issue with sending
  //request itself because of encryption/decryption of password.
  response = sm.execute();
  responseBody = response.haveError() ? response.getErrorMessage() : response.getBody();
  status = response.getStatusCode();
} catch(ex) {
  responseBody = ex.getMessage();
  status = '500';
} finally {
```

© 2017 ServiceNow. All rights reserved.
```javascript
requestBody = sm ? sm.getRequestBody():null;
```

RESTResponseV2
The RESTResponseV2 API allows you to use the data returned by an outbound REST message in JavaScript code.

A RESTResponseV2 object is returned by the RESTMessageV2 functions execute() and executeAsync().

You can use this API in scoped applications, or within the global scope.

RESTResponseV2 - getBody()
Get the content of the REST response body.

Use this function when you want to get the request body as text content. Do not use this method when saving the response as a binary attachment. If a RESTMessageV2 object called the saveResponseBodyAsAttachment(...) function, using getBody() on the associated RESTResponseV2 object will cause an error. When saving the response as an attachment, if the outbound REST message fails, call getErrorMessage() on the response to retrieve the body content.

Table 1142: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1143: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The REST response body.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();
var responseBody = response.getBody();
```

RESTResponseV2 - getCookies()
Get all cookies included in the response.

Table 1144: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1145: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>The list of cookies. Iterate through the list to perform operations on each cookie.</td>
</tr>
</tbody>
</table>

```javascript
var cookies = response.getCookies();
var i;
for(i=0;i<cookies.size();i++) {
    gs.print('cookie: ' + cookies.get(i));
}
```

Output:
- JSESSIONID=4135AA97A5D12DA22EF614AA2B0CAFD8.node20; Path=/; Secure;
  HttpOnly
- SABASESSIONID=370152970.36895.0000; path=/

**RESTResponseV2 - getErrorCode()**
Get the numeric error code if there was an error during the REST transaction.

This error code is specific to the ServiceNow platform, it is not an HTTP error code. Provide this error code if you require assistance from ServiceNow Customer Support.

Table 1146: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1147: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>The numeric error code, such as 1 for socket timeout.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();
var errorCode = response.getErrorCode();
```

**RESTResponseV2 - getErrorMessage()**
Get the error message if there was an error during the REST transaction.
### Table 1148: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 1149: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The error message.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();
var errorMsg = response.getErrorMessage();
```

**RESTResponseV2** - **getHeader(String name)**
Get the value for a specified header.

### Table 1150: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the header that you want the value for, such as Set-Cookie.</td>
</tr>
</tbody>
</table>

### Table 1151: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The value of the specified header.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();
var headerVal = response.getHeader("Content-Type");
```

**RESTResponseV2** - **getHeaders()**
Get all headers returned in the REST response and the associated values.
Table 1152: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1153: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>An Object that maps the name of each header to the associated value.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();
var headers = response.getHeaders();
```

RESTResponseV2 - getQueryString()
Get the fully-resolved query sent to the REST endpoint.

This query contains the endpoint URL as well as any values assigned to variables in the REST message. Use this method only with responses to direct requests. This method is not supported for requests sent asynchronously, or requests sent using a MID server.

Table 1154: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1155: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The fully-resolved query.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); //Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();
var queryString = response.getQueryString();
```

RESTResponseV2 - getResponseAttachmentSysid()
Get the sys_id value of the attachment created from the response body content.

If the RESTMessageV2 object associated with this response called the saveResponseBodyAsAttachment(...) function, use getResponseAttachmentSysid() to get the
sys_id of the created attachment record. Use this function when you want to perform additional operations with the new attachment record.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1157: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The sys_id of the new attachment record.</td>
</tr>
</tbody>
</table>

RESTResponseV2 - getStatusCode()

Get the numeric HTTP status code returned by the REST provider.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1159: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>The numeric status code returned by the REST provider, such as 200 for a successful response.</td>
</tr>
</tbody>
</table>

var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get");  // Might throw exception if message doesn't exist or not visible due to scope. var response = sm.execute(); var statusCode = response.getStatusCode();

RESTResponseV2 - haveError()

Indicate if there was an error during the REST transaction.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1161: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if there was an error, false if there was no error.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();
var error = response.haveError();
```

`RESTResponseV2.waitForResponse(Number timeoutSecs)`
Set the amount of time the instance waits for a response from the web service provider.
This method overrides the property glide.rest.outbound.ecc_response.timeout for this REST response.

Table 1162: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeoutSecs</td>
<td>Number</td>
<td>The amount of time, in seconds, to wait for this response.</td>
</tr>
</tbody>
</table>

Table 1163: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.RESTMessageV2("Yahoo Finance","get"); // Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.executeAsync();
response.waitForResponse(60);
```

**Direct RESTMessageV2 example**
You can send a REST message directly to the endpoint.

In this example, the script sends a REST message requesting a stock quote and waits for a response.
If there is no response from the web service provider, or if the specified REST message record is unavailable, the script throws an error, handled in this example by the try-catch block.

```javascript
var requestBody;
var responseBody;
var status;
var sm;
try{
    sm = new sn_ws.RESTMessageV2("Yahoo Finance", "get"); // Might throw exception if message doesn't exist or not visible due to scope.
    response = sm.executeAsync();
    response.waitForResponse(60);
} catch (err) {
    console.error(err);
}
```
sm.setBasicAuth("admin","admin");
sm.setStringParameter("symbol", "NOW");
sm.setStringParameterNoEscape("xml_data","<data>test</data>");
sm.setHttpTimeout(10000); //In milliseconds. Wait at most 10 seconds for response from http request.
response = sm.execute();//Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password.
responseBody = response.haveError() ? response.getErrorMessage() : response.getBody();
status = response.getStatusCode();
catch(ex) {
    responseBody = ex.getMessage();
    status = '500';
}
finally {
    requestBody = sm ? sm.getRequestBody():null;
}
gs.info("Request Body: " + requestBody);
gs.info("Response: " + responseBody);
gs.info("HTTP Status: " + status);

**Recordless RESTMessageV2 example**
You can use the RESTMessageV2() constructor with no parameters to define a REST message entirely in the script.

When using this constructor you must provide an endpoint and HTTP method. In this example, the script creates an empty REST message and sets the values needed to insert an incident record.

```
var restMessage = new sn_ws.RESTMessageV2();
restMessage.setBasicAuth("admin", "admin");
restMessage.setHttpMethod("post");
restMessage.setEndpoint("http://<instance>.service-now.com/api/now/table/incident");
restMessage.setRequestBody("{"short_description": "Test incident"}");
var response = restMessage.execute();
```

**Asynchronous RESTMessageV2 example**
This example describes how to send an asynchronous REST message using the RESTMessageV2 API.

You can send a REST message asynchronously. When you send an asynchronous message the instance does not wait for a response before proceeding. You must handle waiting for a response within your code.

```
var requestBody;
var responseBody;
var status;
var sm;
try{
    sm = new sn_ws.RESTMessageV2("Yahoo Finance", "get"); // Might throw exception if message doesn't exist or not visible due to scope.
    sm.setBasicAuth("admin","admin");
    sm.setStringParameter("symbol", "NOW");
    sm.setStringParameterNoEscape("xml_data","<data>test</data>");
    response = sm.executeAsync(); //Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password.
    response.waitForResponse(60); // In seconds. Wait at most 60 seconds to get response from ECC Queue/Mid Server //Might throw exception timing out waiting for response in ECC queue.
```
RESTMessageV2 MID server example

You can send a REST message through a MID Server.

By sending the message through a MID Server, you can access endpoints that are behind a firewall or within a private network. All REST messages sent through a MID Server are asynchronous.

```javascript
var requestBody;
var responseBody;
var status;
var sm;
try{
    sm = new sn_ws.RESTMessageV2("Yahoo Finance", "get");  // Might throw exception if message doesn't exist or not visible due to scope.
    sm.setBasicAuth("admin","admin");
    sm.setStringParameter("symbol", "NOW");
    sm.setStringParameterNoEscape("xml_data","<data>test</data>");
    sm.setMIDServer('mid_server_name');
    response = sm.execute(); // Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password.

    response.waitForResponse(60);// In seconds. Wait at most 60 seconds to get response from ECC Queue/Mid Server //Might throw exception timing out waiting for response in ECC queue.

    responseBody = response.haveError() ? response.getErrorMessage() : response.getBody();
    status = response.getStatusCode();
} catch(ex) {
    responseBody = ex.getMessage();
    status = '500';
} finally {
    requestBody = sm ? sm.getRequestBody():null;
}

gs.info("Request Body: " + requestBody);
gs.info("Response: " + responseBody);
gs.info("HTTP Status: " + status);
```

Outbound SOAP web service

The SOAP Message module can be used to develop, prototype, and save outbound SOAP messages that can be reused in business rules and scripts.

You can use outbound SOAP messages in scripts using the SOAPMessageV2 API and the SOAPResponseV2 API. Examples detailing how to script outbound SOAP are available.
Outbound SOAP video tutorial

The following video tutorial demonstrates how to configure outbound SOAP web service messages to consume third-party web services from an instance.

SOAP message

Information needed to send SOAP requests is stored in SOAP message records.

Information needed to send SOAP requests is stored in SOAP message records. Each record specifies an endpoint for the request, the required format of the request as a web services description language (WSDL) file, authentication information, and a list of functions that can run against the endpoint.

Create a SOAP message

Create a SOAP message to define the remote endpoint, WSDL, and authentication settings.

Role required: web_service_admin

1. Navigate to System Web Services > SOAP Message.
2. Click New.
3. Enter a Name to identify the SOAP message.
4. Specify a WSDL using one of these options:
   - To download and use an online WSDL source, select the Download WSDL check box and enter the URL for the WSDL in the WSDL field.
   - To enter the WSDL directly, clear the Download WSDL check box, and then copy and paste the WSDL XML into the WSDL XML field.
5. If the endpoint is protected by basic authentication, select the Use basic auth check box and enter the credentials.
6. If the endpoint requires mutual authentication, select the Enable mutual authentication check box and select a Protocol profile to use for mutual authentication.
7. Click Submit.

This image shows an example of a SOAP message that connects to a demo instance of ServiceNow.
### SOAP Message

<table>
<thead>
<tr>
<th>Name</th>
<th>Created By</th>
<th>Created</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>demo1 incident</td>
<td>dico</td>
<td>2010-03-31 17:30:58</td>
<td>Sample ServiceNow direct web service</td>
</tr>
</tbody>
</table>

#### Related Links

Generate sample SOAP messages

#### SOAP Message Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>SOAP action</th>
<th>Envelope</th>
<th>Lock</th>
<th>Updated</th>
</tr>
</thead>
</table>

Actions on selected rows...
SOAP message functions

After you create a SOAP message record, you can click Generate sample SOAP messages to populate the SOAP Message Functions related list.

The instance creates these functions by reading the supplied WSDL definition.
Figure 837: Soap message function

```xml
 xmlns:ns1="http://www.service-now.com/">
  <SOAP-ENV:Body>
    <ns1:insert>
      <short_description xsi:type="xsd:string">${short_description}</short_description>
    </ns1:insert>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
The **SOAP action**, **SOAP endpoint**, and **Envelope** fields should be populated automatically based on the WSDL definition. The **Envelope** defines the message to send to the endpoint. In this example, the **Envelope** values have this format:

```xml
... <!-- optional -->
<short_description xsi:type="xsd:string">String</short_description>
... 
```

To submit a specific value, enter the value directly in the appropriate XML tag. In this example, to set the **Short description** for a record, enter:

```xml
...
<short_description xsi:type="xsd:string">This is the short description</short_description>
... 
```

**Variable substitution in outbound SOAP**

To use variable substitution, use the format `${<variable_name>}` instead of defining a specific value.

```xml
...
<short_description xsi:type="xsd:string">${short_desc}</short_description>
... 
```

To test variable substitution after you have modified the SOAP envelope with the variables, define values for the variables in the SOAP Message Parameters related list. For example, click **New** and enter the following information:

![Soap message parameters](Image)

**Figure 838: Soap message parameters**

**SOAP envelope variables**

Variables within the SOAP **Envelope** allow you to set values dynamically when sending the SOAP message.

When a SOAP message is generated, the variable names depend on the XML structure surrounding the variable. By default, the XML elements two levels above the variable determine the variable name.

For example, the structure `<A><B><C>${B.C}</C></B></A>` produces the variable `${B.C}`.

These properties control the behavior of variables in the SOAP envelope.
### Table 1164: Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Type</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.soap.outbound.include_variable_context</td>
<td>Controls if variable names include higher-level XML elements. Set this property to false to only use the lowest-level XML element to determine the variable name, such as ${C} instead of ${B.C}.</td>
<td>true/false</td>
<td>true</td>
</tr>
<tr>
<td>glide.soap.outbound.variable_context_depth</td>
<td>Controls the number of higher-level XML elements to include in variable names. For example, when this property value is set to 3, the structure <code>&lt;A&gt;&lt;B&gt;&lt;C&gt;${A.B.C}&lt;/C&gt;&lt;/B&gt;&lt;/A&gt;</code> produces the variable ${A.B.C}.</td>
<td>integer</td>
<td>2</td>
</tr>
</tbody>
</table>

**Test the SOAP message**

Test a SOAP message to validate the configuration before using the message in an integration.

To test the SOAP message, click the **Test** related link. You are redirected to a test result form as shown below.
Figure 839: Soap message test

You can see the original SOAP request message, the resulting HTTP status code, and the SOAP response in this screen. You can also click the Rerun test related link to resubmit the SOAP request.

**Note:** A test SOAP message will time out after 60 seconds if a response is not received.

*Use a SOAP message in a script*

Preview and reuse the script to send the SOAP message.

After you have developed and tested the SOAP message, click the Preview script usage related link in the SOAP Message Function form. The dialog box displays an example of how you can invoke the SOAP message from a script.
Figure 840: Soap message sample script

You can manipulate the resulting XML response body with `XMLDocument` or with `gs.getXMLText` and `gs.getXMLNodeList`.

Send a SOAP message through a MID server
When creating SOAP message functions, you can configure the function to be sent through a MID Server. There must be a running MID Server associated with your instance to use this functionality. All SOAP messages sent through a MID Server are performed asynchronously.
By specifying a MID Server, all SOAP requests that use this SOAP message are sent through that MID Server. You can override the selected MID Server by using the `setMIDServer(mid server)` API call in a script.

Create a SOAP message from a WSDL that references an external XSD file

Create a SOAP message from a WSDL and external XSD file.

Role required: web_service_admin

This task includes example WSDL and XSD files for a weather forecast SOAP message. Your WSDL and XSD file will vary.

1. Navigate to System Web Services > SOAP Message and create a new record.
2. Clear the Download WSDL check box.
3. Paste the content of the WSDL into the WSDL XML field.
4. Save the record.
5. In the SOAP Message Imports related list, click New.
6. Paste the content of the XSD file into the **External Document** field.
7. Set the **Schema Location** field to `db://<name of the referenced XSD file>.xsd`.
   Specifying the schema location allows the instance to know the location of the referenced XSD file.
8. Click **Submit**.
9. Click **Generate sample SOAP messages**.

**Example WSDL and XSD files**

```xml
<?xml version="1.0" encoding="utf-8"?>
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/
  schemas.xmlsoap.org/wsd1/soap/" xmlns:wssn="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/
  targetNamespace="http://www.webservicex.net"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/
    wsdl/" Get one week weather forecast for valid zip code or Place
  name in USA</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://www.webservicex.net">
      <s:include schemaLocation="WeatherForecast.xsd" />
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetWeatherByZipCodeSoapIn">
    <wsdl:part name="parameters" element="tns:GetWeatherByZipCode" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByZipCodeSoapOut">
    <wsdl:part name="parameters" element="tns:GetWeatherByZipCodeResponse" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByPlaceNameSoapIn">
    <wsdl:part name="parameters" element="tns:GetWeatherByPlaceName" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByPlaceNameSoapOut">
    <wsdl:part name="parameters" element="tns:GetWeatherByPlaceNameResponse" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByZipCodeHttpGetIn">
    <wsdl:part name="ZipCode" type="s:string" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByZipCodeHttpGetOut">
    <wsdl:part name="Body" element="tns:WeatherForecasts" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByPlaceNameHttpGetIn">
    <wsdl:part name="PlaceName" type="s:string" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByPlaceNameHttpGetOut">
    <wsdl:part name="Body" element="tns:WeatherForecasts" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByZipCodeHttpPostIn">
    <wsdl:part name="ZipCode" type="s:string" />
  </wsdl:message>
  <wsdl:message name="GetWeatherByZipCodeHttpPostOut">
    <wsdl:part name="Body" element="tns:WeatherForecasts" />
  </wsdl:message>
</wsdl:definitions>
```
<wsdl:message name="GetWeatherByPlaceNameHttpPostIn">
    <wsdl:part name="PlaceName" type="s:string" />
</wsdl:message>
<wsdl:message name="GetWeatherByPlaceNameHttpPostOut">
    <wsdl:part name="Body" element="tns:WeatherForecasts" />
</wsdl:message>
<wsdl:message name="GetWeatherByPlaceNameHttpPostOut">
    <wsdl:part name="Body" element="tns:WeatherForecasts" />
</wsdl:message>
<wsdl:message name="GetWeatherByPlaceNameHttpPostOut">
    <wsdl:part name="Body" element="tns:WeatherForecasts" />
</wsdl:message>

<wsdl:message name="GetWeatherByPlaceNameHttpPostOut">
    <wsdl:part name="Body" element="tns:WeatherForecasts" />
</wsdl:message>

<wsdl:portType name="WeatherForecastSoap">
    <wsdl:operation name="GetWeatherByZipCode">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/>
        <wsdl:input message="tns:GetWeatherByZipCodeSoapIn" />
        <wsdl:output message="tns:GetWeatherByZipCodeSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetWeatherByPlaceName">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/>
        <wsdl:input message="tns:GetWeatherByPlaceNameSoapIn" />
        <wsdl:output message="tns:GetWeatherByPlaceNameSoapOut" />
    </wsdl:operation>
</wsdl:portType>

<wsdl:portType name="WeatherForecastHttpGet">
    <wsdl:operation name="GetWeatherByZipCode">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/>
        <wsdl:input message="tns:GetWeatherByZipCodeHttpGetIn" />
        <wsdl:output message="tns:GetWeatherByZipCodeHttpGetOut" />
    </wsdl:operation>
    <wsdl:operation name="GetWeatherByPlaceName">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/>
        <wsdl:input message="tns:GetWeatherByPlaceNameHttpGetIn" />
        <wsdl:output message="tns:GetWeatherByPlaceNameHttpGetOut" />
    </wsdl:operation>
</wsdl:portType>

<wsdl:portType name="WeatherForecastHttpPost">
    <wsdl:operation name="GetWeatherByZipCode">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/>
        <wsdl:input message="tns:GetWeatherByZipCodeHttpPostIn" />
        <wsdl:output message="tns:GetWeatherByZipCodeHttpPostOut" />
    </wsdl:operation>
    <wsdl:operation name="GetWeatherByPlaceName">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/>
        <wsdl:input message="tns:GetWeatherByPlaceNameHttpPostIn" />
        <wsdl:output message="tns:GetWeatherByPlaceNameHttpPostOut" />
    </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="WeatherForecastSoap" type="tns:WeatherForecastSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="GetWeatherByZipCode">
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetWeatherByPlaceName">
  <soap:operation soapAction="http://www.webservicex.net/GetWeatherByPlaceName" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:binding name="WeatherForecastSoap12" type="tns:WeatherForecastSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="GetWeatherByZipCode">
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetWeatherByPlaceName">
    <soap12:operation soapAction="http://www.webservicex.net/GetWeatherByPlaceName" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

<wsdl:binding name="WeatherForecastHttpGet" type="tns:WeatherForecastHttpGet">
  <http:binding verb="GET" />
  <wsdl:operation name="GetWeatherByZipCode">
    <http:operation location="/GetWeatherByZipCode" />
    <wsdl:input>
      <http:uriEncoded />=3446
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml part="Body" />=3446
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetWeatherByPlaceName">
    <http:operation location="/GetWeatherByPlaceName" />
    <wsdl:input>
      <http:uriEncoded />=3446
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml part="Body" />=3446
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<s:schema elementFormDefault="qualified" targetNamespace="http://www.webservicex.net"
  <s:element name="GetWeatherByZipCode">
    <s:complexType>
      <s:sequence>
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
<s:element minOccurs="0" maxOccurs="1" name="ZipCode" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="GetWeatherByZipCodeResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="GetWeatherByZipCodeResult" type="tns:WeatherForecasts" />
</s:sequence>
</s:complexType>
</s:element>
<s:complexType name="WeatherForecasts">
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="Latitude" type="s:float" />
<s:element minOccurs="1" maxOccurs="1" name="Longitude" type="s:float" />
<s:element minOccurs="1" maxOccurs="1" name="AllocationFactor" type="s:float" />
<s:element minOccurs="0" maxOccurs="1" name="FipsCode" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="PlaceName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="StateCode" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Status" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Details" type="tns:ArrayOfWeatherData" />
</s:sequence>
</s:complexType>
</s:complexType>
<s:complexType name="ArrayOfWeatherData">
<s:sequence>
<s:element minOccurs="0" maxOccurs="unbounded" name="WeatherData" type="tns:WeatherData" />
</s:sequence>
</s:complexType>
<s:complexType name="WeatherData">
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Day" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="WeatherImage" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="MaxTemperatureF" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="MinTemperatureF" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="MaxTemperatureC" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="MinTemperatureC" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="GetWeatherByPlaceName">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="PlaceName" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="GetWeatherByPlaceNameResponse"/>
Demonstration video
This video shows examples of creating and sending SOAP messages.
The script example in this video uses the version one SOAPMessage API.

Connectivity details
For the ServiceNow-initiated SOAP requests to successfully communicate with the web service provider inside a remote network, the ServiceNow instance must have HTTP or HTTPS access to the SOAP endpoint at the provider.

Like any integration, such as LDAP, web services, or JDBC, the SOAP endpoint may reside behind a firewall that is blocking inbound communication from the instance. If this is the case, you need to make network changes to allow this connectivity into your network. You can either modify the firewall and ACL rules to allow the instance IP address, configure the SOAP message to use a MID Server, or implement a VPN tunnel to allow the instance communication into your network.

Note: A common misconception is that because asynchronous SOAP requests are routed through the ECC queue, they are always sent through a MID Server. This is not the case. Asynchronous SOAP requests only use a MID Server when configured to do so.

Outbound SOAP security
You can authenticate outbound SOAP messages using several different security protocols.
The security protocol you should use depends on the requirements of the web service provider. Mutual authentication is supported for outbound web services.

Basic authentication
If the endpoint requires a user name and password, you can provide credentials using basic authentication.

1. Navigate to System Web Services > Outbound > SOAP Message.
2. Select a SOAP message record.
3. In the SOAP Message Functions related list, select a function.
4. Select Use basic auth.
5. Enter a user name in the Basic auth user ID field.
6. Enter the password for that user in Basic auth user password.
7. Click Update.

Set web service security
You can sign outbound SOAP messages using a key store and trusted server certificate saved on the instance.
Role required: admin

1. Upload a key store certificate with a Type of Java Key Store or PKCS12 Key Store.
2. Upload the trusted server certificate for the key store certificate.
4. Select a SOAP message record.
5. In the SOAP Message Functions related list, select a function.
7. In the Key store field, select the Java or PKCS12 key store you uploaded.
8. In the Key store alias field, enter the alias that identifies the public and private key pair.
9. In the Key store password field, enter the password you assigned the key store record.
10. In the Certificate field, select the trusted certificate for the selected key store.
11. Click Update.

Mutual authentication
ServiceNow supports mutual authentication for outbound web services.

Mutual authentication is not available for inbound web services.

Configure SOAP with a proxy

Certain properties provide support for SOAP requests to use a web proxy server.

Table 1165: SOAP properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.proxy_host</td>
<td>The proxy server hostname or IP address</td>
<td>proxy.company.com, 192.168.34.54</td>
</tr>
<tr>
<td>glide.http.proxy_port</td>
<td>The port number for the proxy server</td>
<td>8080, 9100</td>
</tr>
<tr>
<td>glide.http.proxy_username</td>
<td>If the proxy server is authenticating using user name and password, enter a value for this property</td>
<td>proxyuser</td>
</tr>
<tr>
<td>glide.http.proxy_password</td>
<td>If the proxy server is authenticating using user name and password, enter a value for this property</td>
<td>password</td>
</tr>
</tbody>
</table>

SOAPMessageV2

The SOAPMessageV2 API allows you to send an outbound SOAP message using JavaScript.

Use the SOAPResponseV2 API to manage the response returned by the SOAP provider.

You can use this API in scoped applications, or within the global scope.

SOAPMessageV2 - execute()
Send the SOAP message to the endpoint.
Table 1166: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1167: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPResponseV2 on page 3464</td>
<td>The response returned by the SOAP provider.</td>
</tr>
</tbody>
</table>

var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote");  // Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.execute();  // Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password.

SOAPMessageV2 - executeAsync()
Send the SOAP message to the ECC queue.

SOAP messages in the ECC queue are processed by the SOAPClient business rule.

By default, this business rule does not run asynchronously. To configure this business rule to run asynchronously, set the **When** value to Async and add current.update() to the end of the **Script**. The instance does not wait for a response from the web service provider when sending a message through the ECC queue.

Table 1168: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1169: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPResponseV2 on page 3464</td>
<td>The response returned by the SOAP provider.</td>
</tr>
</tbody>
</table>

**Note:** Attempting to use the SOAP response object before the response has been processed may result in a timeout error.

var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote");  // Might throw exception if message doesn't exist or not visible due to scope.
var response = sm.executeAsync();

`SOAPMessageV2.getEndpoint()`  
Get the endpoint for the SOAP message.

**Table 1170: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1171: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The URL of the SOAP web service provider.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
var endpoint = sm.getEndpoint();
```

`SOAPMessageV2.getRequestBody()`  
Get the content of the SOAP message body.

**Table 1172: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1173: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The SOAP message body.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
var endpoint = sm.getRequestBody();
```

`SOAPMessageV2.getRequestHeader(String headerName)`  
Get the value for an HTTP header specified by the SOAP client.

By default, this method cannot return the value for a header set automatically by the system. To grant this method access to all headers, set the property glide.http.log_debug to true.
### Table 1174: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>headerName</td>
<td>String</td>
<td>The request header you want to get the value for.</td>
</tr>
</tbody>
</table>

### Table 1175: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The value of the specified header.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
var header = sm.getRequestHeader("Accept");
```

#### SOAPMessageV2 - getRequestHeaders()

Get HTTP headers that were set by the SOAP client and the associated values.

This method does not return headers set automatically by the system. To configure this method to return all headers, set the property glide.http.log_debug to true.

### Table 1176: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 1177: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>An Object that maps the name of each header to the associated value.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
var endpoint = sm.getRequestHeaders();
```

#### SOAPMessageV2 - setBasicAuth(String userName, String userPass)

Set basic authentication headers for the SOAP message.

Setting basic authentication headers using this method overrides basic authentication values defined in the SOAP message record.
### Table 1178: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>String</td>
<td>The username to use when authenticating the SOAP message.</td>
</tr>
<tr>
<td>userPass</td>
<td>String</td>
<td>The password for the specified user.</td>
</tr>
</tbody>
</table>

### Table 1179: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote");  // Might throw exception if message doesn't exist or not visible due to scope.
sm.setBasicAuth("username","password");
```

**SOAPMessageV2 - setEccCorrelator(String correlator)**

Associate outbound requests and the resulting response record in the ECC queue.

This method only applies to SOAP messages sent through a MID Server. The correlator provided populates the **Agent correlator** field on the ECC queue record for the response. Provide a unique correlator for each outbound request to associate the correct results in the ECC queue with the request when designing asynchronous automation through a MID Server.

### Table 1180: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlator</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
</tbody>
</table>

### Table 1181: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote");  // Might throw exception if message doesn't exist or not visible due to scope.
sm.setEccCorrelator("unique_id");
```

**SOAPMessageV2 - setEccParameter(String name, String value)**

Override a value from the database by writing to the SOAP message payload.
This method only applies to SOAP messages sent through a MID Server. Use this method when a value from the SOAP message in the database is invalid, such as when the endpoint URL is longer than the maximum SOAP endpoint field length.

These are valid values for the name parameter.

- **source**: the endpoint URL.
- **name**: the SOAP message function to run.

### Table 1182: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the ECC parameter.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign to the specified ECC parameter.</td>
</tr>
</tbody>
</table>

### Table 1183: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote");  
// Might throw exception if message doesn't exist or not visible  
due to scope.  
sm.setEccParameter("source","http://very.long.endpoint");
```

**SOAPMessageV2 - setEndpoint(String endpoint)**

Set the endpoint for the SOAP message.

By default, the SOAP message uses the endpoint specified in the SOAP message record. Use this method to override the default. You must call this method when using the `SOAPMessageV2()` constructor with no parameters.

### Table 1184: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endpoint</td>
<td>String</td>
<td>The URL of the SOAP web service provider you want to interface with.</td>
</tr>
</tbody>
</table>
Table 1185: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2();
sm.setEndpoint("http://web.service.endpoint");
```

SOAPMessageV2 - `setHttpTimeout(Number timeoutMs)`
Set the amount of time the SOAP message waits for a response from the web service provider before the request times out.

Table 1186: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeoutMs</td>
<td>Number</td>
<td>The amount of time to wait for a response from the web service provider, in milliseconds.</td>
</tr>
</tbody>
</table>

Table 1187: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setHttpTimeout(6000);
```

SOAPMessageV2 - `setMIDServer(String midServerName)`
Configure the SOAP message to be sent through a MID Server.
By default, the SOAP message uses the MID Server specified in the SOAP message function record. Use this method to override the default.

Table 1188: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>midServerName</td>
<td>String</td>
<td>The name of the MID Server you want to send the SOAP message through. Your instance must have an active MID Server with the specified name.</td>
</tr>
</tbody>
</table>
### Table 1189: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

**SOAPMessageV2 - setMutualAuth(String profileName)**

Set the mutual authentication protocol profile for the SOAP message.

Setting a protocol profile using this method overrides the protocol profile selected for the SOAP message record.

### Table 1190: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>profileName</td>
<td>String</td>
<td>The name of the protocol profile to use for mutual authentication.</td>
</tr>
</tbody>
</table>

### Table 1191: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote");  // Might throw exception if message doesn't exist or not visible due to scope.
sm.setMutualAuth("auth_profile_name");
```

**SOAPMessageV2 - setRequestBody(String requestBody)**

Set the body content to send to the web service provider.

When you set the body content using this method, variables in the body are not substituted for parameters from the SOAP message function record. You must explicitly define all values within the SOAP message body.

### Table 1192: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestBody</td>
<td>String</td>
<td>The body of the SOAP message.</td>
</tr>
</tbody>
</table>
### Table 1193: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
var body = "<SOAP message body>";
sm.setRequestBody(body);  
```

**SOAPMessageV2 - setRequestHeader(String headerName, String headerValue)**

Set an HTTP header in the SOAP message to the specified value.

### Table 1194: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>headerName</td>
<td>String</td>
<td>The name of the header.</td>
</tr>
<tr>
<td>headerValue</td>
<td>String</td>
<td>The value to assign to the specified header.</td>
</tr>
</tbody>
</table>

### Table 1195: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setRequestHeader("Accept","Application/json");  
```

**SOAPMessageV2 - setSOAPAction(String soapAction)**

Define the SOAP action this SOAP message performs.

The WSDL for your web service provider lists SOAP actions you can perform. You must call this method when using the `SOAPMessageV2()` constructor with no parameters.

### Table 1196: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>soapAction</td>
<td>String</td>
<td>The SOAP action this SOAP message performs.</td>
</tr>
</tbody>
</table>
Table 1197: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.SOAPMessageV2();
sm.setSOAPAction("GetQuote");
//construct SOAP message by specifying endpoint and auth
sm.execute();
```

**SOAPMessageV2 - setStringParameter(String name, String value)**
Set a variable with the specified name from the SOAP message record to the specified value.
XML reserved characters in the value are converted to the equivalent escaped characters.

Table 1198: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the SOAP message variable.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign to the specified variable.</td>
</tr>
</tbody>
</table>

Table 1199: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```java
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
sm.setStringParameter("symbol","NOW");
```

**SOAPMessageV2 - setStringParameterNoEscape(String name, String value)**
Set a variable with the specified name from the SOAP message record to the specified value.
This method is equivalent to setStringParameter but does not escape XML reserved characters.

Table 1200: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the SOAP message variable.</td>
</tr>
</tbody>
</table>
### Table 1201: Returns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
<td>The value to assign to the specified variable.</td>
</tr>
</tbody>
</table>

### Table 1202: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keystoreId</td>
<td>String</td>
<td>The sys_id of the Java or PKCS12 key store to use.</td>
</tr>
<tr>
<td>keystoreAlias</td>
<td>String</td>
<td>The alias that identifies the public and private keys.</td>
</tr>
<tr>
<td>keystorePassword</td>
<td>String</td>
<td>The password assigned to the key store record.</td>
</tr>
<tr>
<td>certificateId</td>
<td>String</td>
<td>The sys_id of the trusted server certificate.</td>
</tr>
</tbody>
</table>

### Table 1203: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>
SOAPMessageV2 - SOAPMessageV2()
Instantiates an empty SOAPMessageV2 object.

When using an object instantiated this way, you must manually specify a SOAP action and endpoint.

### Table 1204: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2();
```

SOAPMessageV2 - SOAPMessageV2(String soapMessage, String soapFunction)
Instantiate a SOAPMessageV2 object from a SOAP message record and a function associated with that record.

Values such as the endpoint, authentication, or MID Server settings from the SOAP message record apply to this object.

### Table 1205: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>soapMessage</td>
<td>String</td>
<td>The SOAP message record you want to use as the base for this object.</td>
</tr>
<tr>
<td>soapFunction</td>
<td>String</td>
<td>The SOAP function you want to execute. Available SOAP functions depend on the WSDL supplied by the web service provider.</td>
</tr>
</tbody>
</table>

```javascript
var sm = new sn_ws.SOAPMessageV2("StockQuote","GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope.
```

**Asyncronous SOAPMessageV2 example**
You can send a SOAP message asynchronously.

When you send an asynchronous message the instance does not wait for a response before proceeding. You must handle waiting for a response within your code.

```javascript
var requestBody;
var responseBody;
```
var status;
var sm;
try{
    sm = new sn_ws.SOAPMessageV2("StockQuote", "GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope
    sm.setBasicAuth("admin","admin");
    sm.setStringParameter("symbol", "NOW");
    sm.setStringParameterNoEscape("xml_data","<data>test</data>");
    response = sm.executeAsync(); // Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password
    response.waitForResource(60); // In Seconds, Wait at most 60 seconds for response from ECC Queue/Mid Server // Might throw exception timing out waiting for response in ECC queue
    responseBody = response.haveError() ? response.getErrorMessage() : response.getBody();
    status = response.getStatusCode();
} catch(ex) {
    responseBody = ex.getMessage();
    status = '500';
} finally {
    requestBody = sm ? sm.getRequestBody():null;
}
gs.info("Request Body: " + requestBody);
gs.info("Response: " + responseBody);
gs.info("HTTP Status: " + status);

Direct SOAPMessageV2 example
You can send a SOAP message directly to the endpoint.

In this example, the script sends a SOAP message requesting a stock quote and waits for a response. If there is no response from the web service provider, or if the specified SOAP message record is unavailable, the script throws an error, handled in this example by the try-catch block.

var requestBody;
var responseBody;
var status;
var sm;
try{
    sm = new sn_ws.SOAPMessageV2("StockQuote", "GetQuote"); // Might throw exception if message doesn't exist or not visible due to scope
    sm.setBasicAuth("admin","admin");
    sm.setStringParameter("symbol", "NOW");
    sm.setStringParameterNoEscape("xml_data","<data>test</data>");
    sm.setHttpTimeout(10000) // In Milli seconds. Wait at most 10 seconds for response from http request.
    response = sm.execute(); // Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password and stuff
    responseBody = response.haveError() ? response.getErrorMessage() : response.getBody();
    status = response.getStatusCode();
} catch(ex) {
    responseBody = ex.getMessage();
    status = '500';
} finally {
    requestBody = sm ? sm.getRequestBody():null;
}
gs.info("Request Body: " + requestBody);
gs.info("Response: " + responseBody);
Recordless SOAPMessageV2 example
You can use the SOAPMessageV2() constructor with no parameters to define a SOAP message entirely in the script.

When using this constructor you must provide an endpoint and SOAP action. In this example, the script creates an empty SOAP message and sets the values needed to insert an incident record.

```javascript
var s = new sn_ws.SOAPMessageV2(); //create an empty SOAP message
s.setBasicAuth('admin','admin');
s.setSOAPAction('http://www.service-now.com/incident/insert'); //set the SOAP action to perform
s.setEndpoint('http://<instance>.service-now.com/incident.do?SOAP'); //set the web service provider endpoint
var response = s.execute();
var xmldoc = new XMLDocument(response.getBody());
var incident_sysid = xmldoc.getNodeText('//sys_id');
```

SOAPMessageV2 MID server example
You can send a SOAP message through a MID Server.

By sending the message through a MID Server, you can access endpoints that are behind a firewall or within a private network. All SOAP messages sent through a MID Server are asynchronous.

```javascript
var requestBody;
var responseBody;
var status;
var sm;
try{
    sm = new sn_ws.SOAPMessageV2("StockQuote", "GetQuote");  // Might throw exception if message doesn't exist or not visible due to scope
    sm.setBasicAuth("admin","admin");
    sm.setStringParameter("symbol", "NOW");
    sm.setStringParameterNoEscape("xml_data","<data>test</data>");
    sm.setMIDServer('mid_server_name');
    response = sm.execute();//Might throw exception if http connection timed out or some issue with sending request itself because of encryption/decryption of password and stuff
    response.waitForResponse(60);// In Seconds, Wait at most 60 seconds to get response from ECC Queue/Mid Server //Might throw exception timing out waiting for response in ECC queue

    responseBody = response.haveError() ? response.getErrorMessage() : response.getBody();
    status = response.getStatusCode();
} catch(ex) {
    responseBody = ex.getMessage();
    status = '500';
} finally {
    requestBody = sm ? sm.getRequestBody():null;
} 
gs.info("Request Body: " + requestBody);
gs.info("Response: " + responseBody);
gs.info("HTTP Status: " + status);
```
SOAPResponseV2

The SOAPResponseV2 API allows you to use the data returned by an outbound SOAP message in JavaScript code.

A SOAPResponseV2 object is returned by the SOAPMessageV2 functions execute() and executeAsync(). You can use this API in scoped applications, or within the global scope.

**SOAPResponseV2 - getBody()**

Get the content of the SOAP response body.

```javascript
var body = response.getBody();
```

**Table 1206: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1207: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The SOAP response body.</td>
</tr>
</tbody>
</table>

**SOAPResponseV2 - getErrorCode()**

Get the numeric error code if there was an error during the SOAP transaction. This error code is specific to the ServiceNow platform, it is not an HTTP error code. Provide this error code if you require assistance from ServiceNow Customer Support.

```javascript
var errorCode = response.getErrorCode();
```

**Table 1208: Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1209: Returns**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>The numeric error code, such as 1 for a socket timeout.</td>
</tr>
</tbody>
</table>

```javascript
var errorCode = response.getErrorCode();
```

**SOAPResponseV2 - getErrorMessage()**

Get the error message if there was an error during the SOAP transaction.
Table 1210: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1211: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The error message.</td>
</tr>
</tbody>
</table>

```javascript
var errorMsg = response.getErrorMessage();
```

SOAPResponseV2 - `getHeader(String name)`
Get the value for a specified HTTP header.

Table 1212: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the header that you want the value for, such as Set-Cookie.</td>
</tr>
</tbody>
</table>

Table 1213: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The value of the specified header.</td>
</tr>
</tbody>
</table>

```javascript
var headerVal = response.getHeader("Accept");
```

SOAPResponseV2 - `getHeaders()`
Get all HTTP headers returned in the SOAP response and the associated values.

Table 1214: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1215: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>An Object that maps the name of each header to the associated value.</td>
</tr>
</tbody>
</table>

```javascript
var headers = response.getHeaders();
```

**SOAPResponseV2 - getStatusCode()**
Get the numeric HTTP status code returned by the SOAP provider.

### Table 1216: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 1217: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>The numeric status code returned by the SOAP provider, such as 200 for a successful response.</td>
</tr>
</tbody>
</table>

```javascript
var statusCode = response.getStatusCode();
```

**SOAPResponseV2 - haveError()**
Indicate if there was an error during the SOAP transaction.

### Table 1218: Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 1219: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>Returns true if there was an error, false if there was no error.</td>
</tr>
</tbody>
</table>

```javascript
var error = response.haveError();
```
SOAPResponseV2 - waitForResponse(Number timeoutSecs)
Set the amount of time the instance waits for a response from the web service provider.
This method overrides the property glide.soap.outbound.ecc_response.timeout for this SOAP response.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeoutSecs</td>
<td>Number</td>
<td>The amount of time, in seconds, to wait for this response.</td>
</tr>
</tbody>
</table>

Table 1221: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
</tbody>
</table>

```javascript
response.waitForResponse(60);
```

SOAPResponseV2 - getCookies()
Get all cookies included in the response.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1223: Returns

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>The list of cookies. Iterate through the list to perform operations on each cookie.</td>
</tr>
</tbody>
</table>

```javascript
var cookies = response.getCookies();
var i;
for(i=0;i<cookies.size();i++) {
    gs.print('cookie: ' + cookies.get(i));
}
```

Output:
- JSESSIONID=4135AA97A5D12DA22EF614AA2B0CAFD8.node20; Path=/; Secure; HttpOnly
- SABASESSIONID=370152970.36895.0000; path=/
Outbound web services mutual authentication

Mutual authentication establishes trust by exchanging Secure Sockets Layer (SSL) certificates.

Before connecting to a server, the client requests an SSL certificate. The server responds by requesting that the client send its own certificate. Both respond by validating the certificates of the other and sending acknowledgments before initiating an HTTPS connection. Mutual authentication is not available when making outbound web service calls through a MID Server.

As an administrator, you can enable mutual authentication by defining a protocol profile for connections that require mutual authentication. Protocol profiles allow you to associate a specific certificate record with a protocol, such as HTTPS.

For example, you can create a protocol profile called https for one-way SSL and another called myhttps with a certificate for mutual authentication. You can then make an HTTPS web service request by calling myhttps://<externalendpoint.com> if the end point requires mutual authentication, or https://<externalendpoint.com> if it does not.

Note: This feature enables mutual authentication only on outbound HTTPS connections, such as SOAP, REST, or direct HTTPS calls. ServiceNow does not support mutual authentication for inbound requests or for outbound requests sent through a MID Server.

Create a protocol profile

You can create a protocol profile for outbound web services, such as to enable mutual authentication.

Role required: admin

2. Click New.
3. Fill in the fields on the form, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Enter a unique name to identify this protocol, such as myhttps. The protocol name allows you to differentiate between normal HTTPS connections and HTTPS connections that use this protocol profile.</td>
</tr>
<tr>
<td>Default port</td>
<td>Enter the port number for connections that use this protocol.</td>
</tr>
<tr>
<td>Check box</td>
<td>Select a certificate with a Type value of Java Key Store or PKCS12 Key Store to send to the web service provider. An administrator must set up a key store and generate certificates for the client and server.</td>
</tr>
</tbody>
</table>

Enable mutual authentication

You can configure a SOAP or REST message for mutual authentication using a protocol profile
Role required: web_service_admin or admin

1. Navigate to System Web Services > SOAP Message or System Web Services > REST Message.
2. Select a message record.
3. Select the Use mutual authentication check box.
4. In the Protocol profile field, select a protocol profile configured for mutual authentication.
5. Click Update.

HTTP Connection Management Properties

Connection pooling is controlled by three properties.

The default values for these properties are appropriate for most customers. The Glide properties are dynamic, meaning that changes to these properties will take effect immediately. No outage or restart is required to update the values.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.use_connection_mgr</td>
<td>Switches connection pooling on and off. To disable the new behavior (not recommended) set glide.http.use_connection_mgr to false.</td>
<td>true</td>
</tr>
<tr>
<td>glide.http.connection_mgr.max_connections</td>
<td>Controls the total number of permitted HTTP(S) connections outbound from the ServiceNow ITSA Suite instance. This is an instance-wide value.</td>
<td>20</td>
</tr>
<tr>
<td>glide.http.connection_mgr.max_connections_per_host</td>
<td>Controls how many of the glide.http.connection_mgr.max_connections can communicate in parallel with any particular host. If the maximum setting for any of these values is reached during normal operations, a script or background thread may have to wait briefly to obtain a connection.</td>
<td>4</td>
</tr>
</tbody>
</table>

HTTP Connection Management Properties

Connection pooling is controlled by three properties.

The default values for these properties are appropriate for most customers. The Glide properties are dynamic, meaning that changes to these properties will take effect immediately. No outage or restart is required to update the values.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.use_connection_mgr</td>
<td>Switches connection pooling on and off. To disable the new behavior (not recommended) set glide.http.use_connection_mgr to false.</td>
<td>true</td>
</tr>
</tbody>
</table>
### Property Description Default Value

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.http.connection_mgr.max_connections</td>
<td>Controls the total number of permitted HTTP(S) connections outbound from the ServiceNow ITSA Suite instance. This is an instance-wide value.</td>
<td>20</td>
</tr>
<tr>
<td>glide.http.connection_mgr.max_connections_per_host</td>
<td>Controls how many of the glide.http.connection_mgr.max_connections can communicate in parallel with any particular host. If the maximum setting for any of these values is reached during normal operations, a script or background thread may have to wait briefly to obtain a connection.</td>
<td>4</td>
</tr>
</tbody>
</table>

### ODBC driver

The ServiceNow ODBC driver allows an ODBC client to connect to the ServiceNow platform for reporting. The ServiceNow ODBC driver allows an ODBC client to connect to the ServiceNow platform for reporting. The driver is compliant to version 3.52 of the Microsoft ODBC core API conformance. The ServiceNow ODBC driver uses the ServiceNow web services support for a query-only interface. The ODBC driver supports only **SELECT** statements or read-only functions, and does not modify your instance data. Because the ODBC driver uses the web services interface, platform-wide access control (ACL) is enforced and data security is in place.

This video demonstrates how to configure Microsoft SQL Linked Server with the ODBC driver.

This video demonstrates how to troubleshoot Microsoft SQL Linked Server permissions running out-of-process.

The ODBC driver has these limitations:

- The ODBC driver supports only **SELECT** statements or read-only functions, and does not modify your instance data.
- There is no supported way to use the ODBC driver with a Java client application or with a Java JDBC-ODBC bridge.

**Note:** Versions older than 1.0.7.3 of the ODBC Driver are no longer supported.

### Getting started with ODBC

The ServiceNow Open Database Connectivity (ODBC) driver provides read-only access to the database associated with your ServiceNow instance.

A ServiceNow user with the admin role can perform these procedures.

This video demonstrates how to install, configure, and test the ServiceNow **ODBC driver** on page 3470, which provides read-only access to the database associated with your ServiceNow instance.

Before downloading and installing the ODBC driver, **review the requirements** to ensure your configuration is compatible.
Create an ODBC user - getting started

The ODBC driver communicates with your ServiceNow instance as a specific user. Create a user account for the ODBC driver to use.

1. Navigate to **User Administration > Users**.
2. Click **New**.
3. In the User ID field, enter `odbc.user`.
4. Enter a Password for this user.
5. Click **Submit**.
6. In a separate browser session, confirm that this user is able to log in to your ServiceNow instance.

Set a user permission

Since the ODBC driver communicates with the instance using SOAP, the user account must have the `soap_query` role.

The soap role also provides query access through SOAP, but allows additional actions that are not available through ODBC, such as insert or update.

The user account must also be able to read records on the tables you want to query. For example, to query incident records, the ODBC user must have read permission for the Incident table. You can create a new role, for example a role called ODBC, grant that role to the ODBC user, and configure ACL rules to allow users with that role to query the Incident table.

**Create the ODBC role and grant the required roles**
How to create the ODBC role and grant the ODBC user the required roles.

1. Navigate to **User Administration > Roles**.
2. Click **New**.
3. In the Name field, enter `ODBC`.
4. Click **Submit**.
5. Navigate to **User Administration > Users**.
6. Select the ODBC user account you created.
7. In the Roles related list, click **Edit**.
8. Use the slushbucket to add the ODBC and soap_query roles.

**Define an ACL rule for the ODBC role**
How to define ACL rules for the ODBC role.

1. Elevate the session permissions so you can create ACL rules.
2. Navigate to **System Security > Access Controls (ACL)**.
3. Click **New**.
4. From the Operation choice list, select **read**.
5. From the Name choice list, select `Incident [incident]`.
   Leave the second Name choice list as None.
6. Right-click the form header and select **Save**.
7. In the Requires role related list, click **Edit**.
8. Use the slushbucket to add the ODBC role.
Install the ODBC driver

You must have administrator-level access for the Windows computer onto which you want to install the ODBC driver.

1. Download the ODBC driver version compatible with your computer's operating system and the application you are using to query the database.
   
   The ODBC driver is available as 32-bit or 64-bit. Most applications require the 32-bit ODBC driver even if the operating system is 64-bit.

2. Right-click the executable you downloaded and select Run as Administrator to start the setup wizard.

3. Follow the instructions on the setup wizard. The default values are usually appropriate.

Configure the ODBC driver - getting started

Configure the ODBC driver using the ODBC management console.

1. In Windows, navigate to Start > Programs > ServiceNow ODBC Management Console.

2. Expand the Console Root tree to: OpenAccess SDK 6.0 Manager\<installation location>\Services\ServiceNow_ODBC\Data Source Settings\ServiceNow\IP Parameters

3. Double-click the DataSourceIPProperties attribute.

4. Change the Value to the URL of your ServiceNow instance, such as https://<instance>.service-now.com. If integrating the ODBC driver with Edge Encryption, change the Value to the URL of your encryption proxy. See Integrate Edge Encryption with the ODBC driver for more information.

5. Click OK.

Test the ODBC driver - getting started

Use the ODBC Administrator application to confirm you are able to connect to the instance.

Role required: admin

1. In Windows, navigate to Start > Programs > ServiceNow ODBC > ODBC Administrator.

2. Select the System DSN tab, and then select the ServiceNow data source.

3. Click Configure.

4. Click Test Connect in the ODBC Driver Setup dialog box.

5. Enter odbc.user as the Data Source User Name and the ODBC user's password as the Data Source Password.

6. Click OK.

   The application displays a screen to indicate if the connection was successful.

Installing the ODBC driver

Review setup requirements, download the ODBC driver installer, and install the driver to a computer.

You can install the ServiceNow ODBC driver on Microsoft Windows computers. To install the ODBC driver, set up an ODBC user in your ServiceNow instance, then download and install the ODBC driver. If you already have the ODBC driver installed, you can upgrade to the newest version.

ODBC driver installation requirements

You can install the ServiceNow ODBC driver on Microsoft Windows computers.
To install the ODBC driver, set up an ODBC user in your ServiceNow instance, then download and install the ODBC driver. If you already have the ODBC driver installed, you can upgrade to the newest version.

Requirements

Ensure your configuration meets these requirements before installing the ODBC driver.

Table 1225: Requirements

<table>
<thead>
<tr>
<th>Category</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active user</td>
<td>The user record on the instance used to perform the queries.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The account used to connect to the instance by the ODBC driver must be defined on the instance. Accounts using single sign-on are not supported by the ODBC driver.</td>
</tr>
<tr>
<td>The soap_query role</td>
<td>The user you use to query the database must have the soap_query role if the instance uses the glide.soap.strict_security high security setting.</td>
</tr>
<tr>
<td></td>
<td><strong>Warning:</strong> Do not enable WS-Security for all SOAP requests by setting the glide.soap.require_ws_security system property. It is incompatible with the ODBC driver. Enabling this setting blocks both ODBC driver and MID Server connections. Instead, use basic authentication.</td>
</tr>
<tr>
<td>Target Table ACLs</td>
<td>The user you use to query the database must have read access for the tables that you want to query.</td>
</tr>
<tr>
<td>Target Table Web Service Access</td>
<td>The table you want to query must allow web service interaction. You can enable web service interaction using the application access settings.</td>
</tr>
<tr>
<td>Operating System</td>
<td>The ServiceNow ODBC driver supports installation on the following operating systems:</td>
</tr>
<tr>
<td></td>
<td>• Windows XP</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2003</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2008</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2012</td>
</tr>
<tr>
<td></td>
<td>• Windows Vista</td>
</tr>
<tr>
<td></td>
<td>• Windows 7</td>
</tr>
<tr>
<td></td>
<td>• Windows 8</td>
</tr>
</tbody>
</table>
### Category

**Hardware**
- RAM: 1 GB minimum
- Disk space: 135 MB for installation. 200 MB for writing cache files during usage.

**Account**
The Windows account used for the installation must have local Administrator rights to install an ODBC driver.

**Networking**
During usage, the ODBC driver requires HTTPS (port 443) connectivity to the ServiceNow instance. The communication between the ODBC driver and the ServiceNow instance uses standard SOAP web services.

**End User License Agreement**
Read the End User License Agreement for the ServiceNow ODBC driver.

---

**ODBC driver supported software**
The following table lists the operating systems and reporting applications compatible with each version of the ODBC driver.

<table>
<thead>
<tr>
<th>Driver Version</th>
<th>Operating System</th>
<th>Microsoft Excel</th>
<th>Microsoft SQL Server</th>
<th>Crystal Reports</th>
<th>Tableau</th>
<th>Informatica</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.9 and later</td>
<td>• Windows XP SP2</td>
<td>2007</td>
<td>2008</td>
<td>2008</td>
<td>2008</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>• Windows Vista</td>
<td>2010</td>
<td>2012</td>
<td>2011</td>
<td>2011</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>• Windows 7</td>
<td>2013</td>
<td>2014</td>
<td>2013</td>
<td>2013</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td>• Windows 8.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9.0</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2008 R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2012 R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The ODBC driver provides only basic support for Informatica. Use the ODBC driver with Informatica only for simple operations. Thoroughly test integrations with Informatica before using them in a production environment.
Create an ODBC user

All ODBC queries must be performed as a ServiceNow user. This user must have the soap role and any other roles that are required to read the tables you want to query. An administrator can create a user within the ServiceNow instance for ODBC queries:

1. Create a new user.
2. Give the new user the soap_query role.
3. Give the user any roles needed to view records on the tables you want to query.
   - You may need to create a role with the appropriate ACLs if one does not already exist.
   - It is good practice to grant this user the itil role when querying Task tables.

Downloading the ODBC driver

The ODBC driver is available from the ServiceNow Knowledge Base.

To download the ODBC driver, navigate to [KB0540707](#).

If you do not have access to the Knowledge Base, contact your ServiceNow administrator.

If this is the first time the driver is installed, the installer will be in first time installation mode and prompt for the driver to be installed. Install only one version of the ODBC Driver on a computer. If the ODBC driver was previously installed, the installer will be in upgrade mode and prompt for removal of the previous driver first.

**Note:** Versions older than 1.0.7.3 of the ODBC Driver are no longer supported.

Run ODBC driver setup

Install the ODBC driver for the first time.

1. Right-click the executable and select **Run as Administrator** to launch the installer.
You are presented with the following InstallShield dialog box.

2. Click **Next**.
3. Read and accept the End User License Agreement.
4. Select the target directory for installing the ServiceNow ODBC driver.
   The default directory is C:\Program Files\Service-now\ODBC.
5. Specify the following parameters, which are required to create an ODBC data source that can be used to create a DSN.
   - **Data Source Name**: a short name to identify this data source.
   - **Description**: a short description of the driver. The driver's version number is appended at the end of this value.
   - **Service Name**: the name that can be selected in the **Service Name** field of the ODBC Administrator.
   - **Service Data Source**: the name that can be selected in the **Service Data Source** field of the ODBC Administrator.

   Usually the default values are appropriate.
6. Select the **Program Folder** to create links for the driver. This is the program folder that appears under the **Start** menu.
The installation creates the following links in the menu.

- **Interactive SQL (ODBC):** an interactive SQL command window for directly testing SQL statements.
- **Management Console:** a Microsoft MMC snap-in for configuring default properties for the ODBC driver.
- **ODBC Administrator:** a Microsoft ODBC Administrator program.

The driver code is copied to the target folder.
A progress bar appears.

7. When prompted, click Finish to complete the installation.

**Upgrading the ODBC driver**

If you have previously installed an older version of the ODBC driver, run the installer to uninstall the previous version, and then run the installer again to upgrade.

To check the build date and time of the ODBC driver, use **CheckVersion** located in the Service-Now \ODBC\ip\tools folder. This is an executable Windows host script that reports the build date and time of the current ODBC driver. Use it to assist ServiceNow Technical Support to determine which build of the ODBC driver is running. If the **CheckVersion** tool is absent, the ODBC driver is out of date; upgrade to the current version. To check the version of an older ODBC driver, see the previous version information.

---

**Note:** The ODBC installation also has a Service-Now\ODBC\tools folder, which is not the correct path for the CheckVersion tool.

1. Right-click the executable and select **Run as Administrator**.
2. Click **OK** when prompted to uninstall the current driver, which is required for the upgrade.

A list appears, displaying the existing ODBC DSN names that you have previously created. You have the option to delete them.

3. Select **Yes** to remove all previous DSNs or **No** to keep them for use with the upgraded driver.

An ODBC DSN is a connection *handle* to use the ODBC driver in an application. For more information from Microsoft, see:

4. After removing the previous ODBC driver, double-click the executable again to run the installer. Then, follow the steps in *Installing the ODBC driver*. If you encounter errors when uninstalling the ODBC driver, refer to the *troubleshooting uninstalling ODBC* knowledge article.

After installing the ODBC driver, configure it to connect to your ServiceNow instance.

**Configure the ODBC driver**

After installing the ODBC driver, configure it to connect to your ServiceNow instance and to communicate through a proxy server if applicable, and set properties to control ODBC behavior.

After the driver is installed, configure it for your instance. The driver is preconfigured to connect to https://demoodbc.service-now.com using the DSN ServiceNow. There are two ways to configure connectivity for the driver.

- Configure the global default used by all newly created DSNs.
- Configure each new DSN with its own connection.

To integrate the ODBC driver with Edge Encryption, see *Integrate Edge Encryption with the ODBC driver*. 
Configure the global DSN default

Configure the global default used by all newly created DSNs.

A default DSN is preloaded with the ODBC driver installation ServiceNow data source. This preloaded DSN connects using the default connection URL, which is set to https://demo.service-now.com. To change the global default for the instance URL:

1. In Windows, navigate to **Start > Programs > ServiceNow ODBC > Management Console**.
2. Expand the Console Root tree using the following path:
   
   OpenAccess SDK 6.0 Manager\<installation location>\Services\ServiceNow_ODBC \Data Source Settings\ServiceNow\IP Parameters

3. Double-click the **DataSourceIProperties** attribute for the ServiceNow data source setting to open the Properties dialog box.
4. Change the value to the URL of your instance, using the following format, and then click **OK**:
   
   https://<your instance>.service-now.com

© 2017 ServiceNow. All rights reserved. 3482
Create a new DSN

Using the ODBC driver and the ServiceNow data source, you can create an unlimited number of DSNs configured to connect with different instance URLs.

This allows the flexibility of selecting the target instance for your ODBC connection by DSN name. As an option during installation or upgrade, you can elect to keep the DSNs when you uninstall.

1. In Windows, navigate to Start > Programs > Service-now ODBC > ODBC Administrator.
2. To create a system DSN, select the **System DSN** tab, and then click **Add**.
3. Select ServiceNow ODBC driver 32-bit from the list, and then click Finish.

4. Configure the driver and its connection URL by specifying the url= parameter value in the Custom Properties field. For example:

   url=https://myinstance.service-now.com
5. Click OK.

You can now use the new driver.

Using a connection string

You can specify a connection string instead of defining a DSN.

The connection string must follow this format:

```
Driver=ServiceNow ODBC driver 32-bit;ServiceName=ServiceNow_ODBC;UID=youruser;PWD=yourpassword;ServerDataSource=ServiceNow <instance>.service-now.com
```

The driver name varies depending on if you use the 32-bit or 64-bit version of the ODBC driver. To determine your driver name:

1. In Windows, navigate to Start > Programs > Service-now ODBC > ODBC Administrator.
2. Select the System DSN tab.
3. Note value in the Driver column for the ServiceNow data source.

Configure the logging level of the ODBC driver

You can configure the logging level of the ODBC driver.

1. In Windows, navigate to Start > Programs > ServiceNow ODBC > Management Console.
2. [ODBC version 1.0.8] Within the management console, navigate to `<your_installation_directory> > Services > ServiceNow_ODBC > Service Settings > IP Parameters.

3. [ODBC version 1.0.8] Change the value of the `ServiceJVMOptions` attribute to the desired logging level.

4. Within the management console, navigate to `<your_installation_directory> > Services > ServiceNow_ODBC > Service Settings > Logging.

5. Change the value of the `ServiceDebugLogLevel` by selecting all available check boxes.
6. In Windows, navigate to **Start > Programs > ServiceNow ODBC > ODBC Administrator**.
7. In the ODBC Administrator, select the **Tracing** tab.
8. Navigate to the path in the **Log File Path** field and delete the old log file, if it exists.
9. Click Start Tracing Now.
10. Enable SOAP debugging for your ServiceNow instance.

**ODBC via proxy servers**

The ODBC driver can be configured to route its HTTP SOAP requests via an HTTP proxy server. Setting up a proxy server gives you the option to control access to the ServiceNow instance from the proxy server, and potentially allows a network configuration that can monitor usage statistics. However, because the proxy server intercepts the ODBC driver's requests to your ServiceNow instance, it will degrade the performance of the driver.

**Note:** This feature is recommended for use with ODBC driver builds dated 7/15/2011 or later.

To enable the use of proxy servers, the custom properties for proxy server settings must be defined first for the data source. After that, these properties can be overridden by specific ODBC DSNs. To do this, run the ODBC Management Console.
Figure 842: ODBC proxy

The following custom properties configure the ODBC proxy server.

**Table 1227: Configure ODBC to use proxy servers**

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy_host</td>
<td>The proxy server host name or IP address.</td>
<td>proxy.company.com</td>
</tr>
<tr>
<td>proxy_port</td>
<td>The proxy server port number.</td>
<td>8080</td>
</tr>
<tr>
<td>proxy_user_name</td>
<td>The proxy server user name or id, used in an authenticating proxy configuration.</td>
<td>odbc_user</td>
</tr>
</tbody>
</table>
### Setting ODBC properties

The following properties customize connectivity and optimize the query behavior of the ODBC driver.

#### ODBC Administrator Properties

These properties are specified in the ODBC Data Source Administrator for the DSN or in the **Custom Properties** field of the login dialog box.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BatchSize</strong></td>
<td>During fetching of results from the instance, this batch size configures the number of records to fetch for every request. Typically, the default is an optimal number for normal sized rows. If an error occurs during fetching of records that indicates this value should be lowered, you can modify it to optimize memory usage versus performance.</td>
<td>2000</td>
</tr>
<tr>
<td><strong>url</strong></td>
<td>This is the ServiceNow instance URL or endpoint. It should indicate the URL to the ServiceNow instance you want to connect to.</td>
<td><a href="https://demo.service-now.com">https://demo.service-now.com</a></td>
</tr>
<tr>
<td><strong>EnablePassThrough</strong></td>
<td>During processing of aggregate functions, enabling pass through mode allows directly calling Aggregate Web Service for optimized and speedy response. Whenever possible, this mode should be left enabled.</td>
<td>true</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>debug</td>
<td>By default, debugging messages are not produced. Set <code>debug</code> to <code>true</code> when you operate the ODBC driver from the ISQL console window to write all HTTP-related network communication traffic to the console window. When using this option, set <code>gzip</code> to <code>false</code> so that data is not compressed. Otherwise, the data is unreadable.</td>
<td><code>false</code></td>
</tr>
<tr>
<td>gzip</td>
<td>By default, data sent over the network is compressed. Set <code>gzip</code> to <code>false</code> when using the <code>debug</code> parameter to write network communication to the ISQL console so that data is not compressed.</td>
<td><code>true</code></td>
</tr>
<tr>
<td>timeout</td>
<td>Specifies the socket inactivity timeout value in seconds.</td>
<td>175</td>
</tr>
<tr>
<td>retries</td>
<td>Number of times to retry the failing request in the event of a socket timeout error.</td>
<td>0</td>
</tr>
<tr>
<td>mode</td>
<td>The query mode used to parse complex where clauses. You can configure the ODBC driver query mode to use either AND or OR operators. While the OR operator provides the greatest compatibility with complex queries, the AND operator is usually more efficient and results in fewer database operations.</td>
<td><code>or</code></td>
</tr>
</tbody>
</table>

If you need to use more than one of these properties in your connection, concatenate the settings with a semicolon (:) delimiter. For example, the following string sets the URL to a specific instance and changes the batch size to 200 records.

```
url=https://demo1234.service-now.com;BatchSize=200
```

**ODBC management console properties**

You can access these properties from the ODBC Management Console available in the Windows Start menu at **ServiceNow ODBC > Management Console**.
### Table 1229: ODBC Management Console Properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceJVMOptions (Service\Service Settings\IP Parameters)</td>
<td>JVM command line properties and option. For example, to change the maximum Java heap size, modify the -Xmx150m parameter.</td>
<td>-Xms64m -Xmx150m</td>
</tr>
<tr>
<td>DataSourceIPProperties (OpenAccess SDK 6.0 Manager&lt;installation location&gt;\Services\ServiceNow_ODBC\Data Source Settings\ServiceNow\IP Parameters)</td>
<td>Global default of the instance URL for all ODBC connections. For more flexibility, you may also create new DSNs with default URL configurations.</td>
<td><a href="https://demo.service-now.com">https://demo.service-now.com</a></td>
</tr>
</tbody>
</table>

### Service JVM options

You can specify these values within the ServiceJVMOptions parameter in addition to standard JVM arguments such as -Xmx.

### Table 1230: Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>-DLOG_FILE_NAME</td>
<td>The location of the ODBC log file. This property is available starting with the ODBC driver 1.0.7.1 release.</td>
<td>${user.home}\AppData\Local\ServiceNow\odbc\logging\odbc.log</td>
</tr>
<tr>
<td>-DLOG_LEVEL</td>
<td>The logging level used when writing to the ODBC log file. You can specify the logging level using Logback levels, such as TRACE, INFO, or ERROR. This property is available starting with the ODBC driver 1.0.8 release.</td>
<td>INFO</td>
</tr>
</tbody>
</table>

### Instance properties

An administrator can configure these properties by adding a property or modifying an existing one in the ServiceNow instance.

### Table 1231: Instance Properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.db.max.aggregates</td>
<td>The maximum number of rows returned by aggregate functions.</td>
<td>100000</td>
</tr>
</tbody>
</table>
### Test the ODBC driver

After configuring the ODBC driver, test that the driver can connect to the ServiceNow ITSA Suite instance as the ODBC user and can query data from a target table.

To test the connection, run the ODBC Administrator program.

1. In Windows, navigate to **Start > Programs > ServiceNow ODBC > ODBC Administrator**. The ServiceNow ODBC data source is installed as a system data source.
2. Select the **System DSN** tab, and then select the **ServiceNow** data source.
3. Click **Configure**.

   ![ODBC Data Source Administrator](image)

4. Click **Test Connect** in the ODBC driver Setup dialog box.
5. Enter the login credentials.
These are the usual ServiceNow ITSA SuiteLogin credentials for the ODBC user you created.
6. Click **OK** to log in to the data source.
7. Click **OK** again when the success message appears.

**Enable debug logging**

If you experience unexpected behavior when using the ODBC driver, you can enable debug logging and generate debug logs to help identify the issue.

Debug logs can be useful when submitting an incident with ServiceNow Customer Support.

When you enable debug logging, note the version and bitness (32 bit or 64 bit) of the installed ODBC driver, the Windows operating system, and the client application you are using with the ODBC driver.

To generate debug logs, follow these steps.

1. Close all active applications that may use the ODBC driver.
2. Navigate to one of these paths, based on your operating system.
   - For Windows 7: `C:\Users\<user_name>\AppData\Local\ServiceNow\odbc\logging`
Geneva ServiceNow ServiceNow Platform

- For Windows XP and earlier: C:\Program Files\ServiceNow\ODBC\%LOCALAPPDATA%\ServiceNow\odbc\logging

3. Delete any existing log data to ensure you only log relevant information.
4. Run a query that produces the unexpected behavior, then immediately close the application and review the log files.

Test a query

To verify that the user has the appropriate permissions to send requests to the instance using ODBC, run a query using Interactive SQL.

For testing, use a query that returns exactly one record, such as a query using the Number value of a record.

1. In the ServiceNow ITSA Suite instance, navigate to Incident > All.
2. Record the Number of an incident record.
3. On the computer where the ODBC driver is installed, navigate to Start > Programs > ServiceNow ODBC > Interactive SQL.
4. Enter connect "odbc.user"*"password"@ServiceNow and press the enter key.
5. Enter the following text, substituting the incident number you recorded.
   ```
   select short_description from incident where number='<incident number>';
   ```
6. Press the enter key.

The instance should respond with the short description of the incident record.

ODBC troubleshooting

Review these troubleshooting resources to resolve issues with the ODBC driver.

For troubleshooting information, see the Knowledge Base articles troubleshooting ODBC driver issues and troubleshooting common ODBC error messages.

Querying table and column names

You can get a list of accessible tables and columns, based on the read ACLs for the querying user.

The following query will return the names of all tables that the querying user has read access for:

```
select * from oa_tables;
```

After you know the name of the table you want to query, you can query the names of all columns that the user has read access for. The querying user must have read access for both the table and the columns.

```
select * from oa_columns where table_name='table_name';
```

**Note:** The `oa_tables` and `oa_columns` tables are internal ODBC tables. These tables are accessible only via the ODBC Driver.

ODBC behavior

After testing the ODBC driver you can use it to query your instance database from a variety of client applications.

ODBC aggregate functions

The ODBC driver supports aggregation functions.
The ODBC driver attempts to download the data and apply the aggregate functions locally. The following aggregate functions are supported by the ODBC driver.

- COUNT
- SUM
- MIN
- MAX
- AVG

Activate the Aggregate Web Services plugin to improve the performance of aggregate queries through the ODBC driver.

**ODBC dates and times**

How the ODBC driver expresses dates and times.

Date and time values returned by the ODBC driver are expressed in the local time zone of the application using the driver, not the ServiceNow instance time zone.

**Field length in SQL queries**

The ODBC driver limits the field length in SQL queries to the maximum length defined by the ServiceNow dictionary entry.

Role required: admin

If the data coming from the ODBC source exceeds the field size of the dictionary entry, ServiceNow truncates the query output to fit the field size. You can increase the field size to avoid truncating data.

1. Increase the maximum length in the dictionary entry for the field in question.
2. Reconnect the ODBC driver to pick up the change.

**Note:** By default, the ODBC driver uses the VARCHAR data type to store query string output. When strings become very large (roughly 16000 characters), the ODBC driver uses the LONGVARCHAR data type instead. It is important to keep in mind, however, that the LONGVARCHAR data type has a more limited set of SQL commands that can be executed on it. For example, it does not support queries using scalar data.

**ODBC display values**

Some examples of how to use and work with ODBC display values are shown here.

**Display values in Choice and Reference columns**

When querying a column of type Choice or Reference, an additional column with the prefix `dv_` is available that contains the display value. For example, you can select `dv_caller_id` to return the `sys_user.name` display value of the reference field from an incident record without making another request to the `sys_user` table.
Display values in filter conditions

Display values can also be used in a filter condition. The ODBC driver optimizes the query condition and processes the filter on the server, for example, querying on the display value of `sys_user` for the `caller_id` field of an incident by using the `dv_caller_id` field name.

```
ISQL> select number, dv_caller_id, caller_id from incident;
number dv_caller_id caller_id
------ ---------- -------
INC0000009 Rick Berzle 5137153cc611227c000bbd1bd8cd2006
INC00000010 Fred Luddy 5137153cc611227c000bbd1bd8cd2005
INC00000011 Don Goodliffe 9ee1b13dc6112271007f9d0efdb9cd0
INC00000012 Don Goodliffe 9ee1b13dc6112271007f9d0efdb9cd0
INC00000013 Joe Employee 681ccaf9c0a8016400b98a6818d57c7
INC00000014 Bow Ruggeri f298d2d2c611227b0106c6be7f154bc8
INC00000015 Fred Luddy 5137153cc611227c000bbd1bd8cd2005
INC00000016 Bow Ruggeri f298d2d2c611227b0106c6be7f154bc8
INC00000017 Joe Employee 681ccaf9c0a8016400b98a6818d57c7
INC00000018 Taylor Vreeland 46bac3d6a9fe1981005f299d479b8669
INC00000019 Fred Luddy 5137153cc611227c000bbd1bd8cd2005
```

Figure 844: Display values in filter conditions

Display values in aggregate queries

Aggregate queries can also take advantage of display values if you specify them in the group by or where clause, for example, grouping on the `caller_id` field of an incident, as well as specifying a filter for it. The query is optimized by passing through to the server.

```
ISQL> select number, dv_caller_id, caller_id from incident where dv_caller_id = 'Fred Luddy';
number dv_caller_id caller_id
------ ---------- -------
INC0000010 Fred Luddy 5137153cc611227c000bbd1bd8cd2005
INC0000015 Fred Luddy 5137153cc611227c000bbd1bd8cd2005
INC0000019 Fred Luddy 5137153cc611227c000bbd1bd8cd2005
INC0000027 Fred Luddy 5137153cc611227c000bbd1bd8cd2005
```
Figure 845: Display values in aggregate queries

ODBC and client applications

See the following pages for examples of how to use the ODBC driver to create data sources from other applications.

Use interactive SQL with ODBC

Run the interactive SQL application for quick verification of connectivity and to test query results without using a full application.

1. In Windows, navigate to Start > Programs > ServiceNow ODBC > Interactive SQL (ODBC).
2. Enter the following command to connect to the ServiceNow ITSA Suite instance. Select the appropriate user credentials in the format: ID*password@DSNName. The password cannot contain special characters.

```
CONNECT odbcuser*password@ServiceNow
```
3. Issue a `SELECT` SQL command, such as:

```
SELECT NUMBER, short_description FROM incident;
```

Make sure to include the semicolon at the end of your query statement. You will be presented with a 'Cont>' prompt otherwise.
Specify the maximum number of rows returned

By default, ServiceNow only returns 100 rows of data with each iSQL query. If you need to return more rows of data, set the maxrows parameter for the iSQL session.

To return all rows set maxrows to zero:

```
maxrows 0
```

To return more than 100 rows set maxrows to a higher value. For example, to return 500 rows:

```
maxrows 500
```

**Note:** If running the Interactive SQL console from a shortcut, you must modify the shortcut Target to include the -maxrows parameter with the desired value.
SQL support
The ODBC driver embeds a third party SQL/ODBC engine from DataDirect, a division of Progress Software.

See the DataDirect SQL Reference for information on proper SQL syntax.

**Note:** The ServiceNow ODBC driver only supports `SELECT` statements. The driver ignores other SQL statements such as `CREATE` and `ALTER`. 
ODBC driver in SQL Server

Use the ServiceNow ODBC driver in SQL Server as a Linked Server.

Using the ODBC driver in SQL Server as a **Linked Server** allows SQL Server to query tables from a ServiceNow instance directly via the ODBC driver. Only use the procedures described with SQL Server 2008 and 2012. Other versions of SQL Server may cause unexpected behavior. If you encounter unexpected behavior, refer to the [troubleshooting linked server Knowledge Base article](#).

**Required Permissions**

Additional information on the required permissions for SQL Server Linked Servers can be found [on the MSDN blog](#).

**Note:** Review this information if you encounter permission errors with SQL Server.

**ODBC SQL Server video tutorials**

Watch video tutorials about configuring and troubleshooting the ODBC driver with a SQL Linked Server.

**Configuring Microsoft SQL Linked Server with the ODBC driver**

**Troubleshooting Microsoft SQL Linked Server permissions**

*Configure SQL Server*

The following example configuration was performed on SQL Server 2008, installed on Windows Server 2008.

1. Right-click the SQL Server Management Studio application and select as **Run as Administrator**.
2. Log in to the database to which you want to link.
3. Right-click **Server Objects > Linked Servers**.
4. Click **New Linked Server**.
5. Enter the following values in the dialog.
   - **Linked server:** SERVICENOW. This is the name of the Linked Server.
   - **Provider:** Microsoft OLE DB Provider for ODBC drivers
   - **Product name:** ServiceNow. This is an identifier. Enter any value that is appropriate.
   - **Data source:** ServiceNow. This is the name of your DSN.
6. Select **Security** from the Select a page list, and then enter the following security values:

1. For a login connection, select **Be made using this security context**.
2. Enter the user name and password for connecting to the ServiceNow instance.
3. Click **OK**.
7. Navigate to Server Objects > Linked Server > Providers and double-click Microsoft OLE DB Provider for ODBC drivers.
8. Select the following options.

- Nested Queries
- Level zero only
- Support 'Like' operator

Tip:
ServiceNow recommends running the third-party provider in the out-of-process mode setting (AllowInProcess=FALSE). If you run the provider in-process (within the same process as SQL Server), then any issues with the provider can affect the SQL Server process, which in turn could result in crashing SQL server.
### Provider Options - Microsoft DLE DB Provider for ODBC Drivers

#### Provider options:

<table>
<thead>
<tr>
<th>Enable</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dynamic parameter</td>
</tr>
<tr>
<td></td>
<td>Nested queries</td>
</tr>
<tr>
<td></td>
<td>Level zero only</td>
</tr>
<tr>
<td></td>
<td>Allow inprocess</td>
</tr>
<tr>
<td></td>
<td>Non transactional updates</td>
</tr>
<tr>
<td></td>
<td>Index as access path</td>
</tr>
<tr>
<td></td>
<td>Disallow adhoc access</td>
</tr>
<tr>
<td></td>
<td>Supports &quot;Like&quot; operator</td>
</tr>
</tbody>
</table>

#### Linked servers using this provider:

- **SERVICENOW**

⚠️ These options are applied to all linked servers that use this provider.
9. Test your connection by selecting the newly created linked server SERVICENOW and selecting Test connection.

10. Execute the following query in a query builder window to retrieve some results.

```sql
SELECT * FROM OPENQUERY (SERVICENOW , 'select number, short_description from incident')
```

Number Precision Errors
You may encounter precision errors querying for decimal or number field values using the OPENQUERY syntax with the ODBC driver. In this case, use the Cast syntax to convert the precision. For example:

```sql
SELECT * FROM OPENQUERY (SERVICENOW , 'select Cast(sys_mod_count as Decimal(38,0)), number, short_description from incident')
```

SQL Server Connection String
To use the ODBC driver directly in SQL Server 2008, specify the connection string in the following format.

```
Dsn=ServiceNow;uid =username;pwd =password
```

**Note:** The latest SQL Server 2008 patches are required for the ability to specify a connection string in the user interface, via the SQL import wizard.

**Using sp_addlinkedserver**

The following example creates a linked server named "ServiceNow ODBC" that uses the Microsoft OLE DB Provider for ODBC (MSDASQL) and the data_source parameter.

```
EXEC sp_addlinkedserver
  @server  = N 'ServiceNow ODBC' ,
  @srvproduct  = N '' ,
  @provider  = N 'MSDASQL' ,
  @datasrc  = N 'ServiceNow';
GO
```

After creating the linked server, you must update its properties to specify the login credentials.

**Use the ODBC driver in Excel**

After installing the ODBC driver and its associated DSN, use it in Excel as a data source provider.

1. In Excel open the **Data** tab.
2. Under **From Other Sources** open **From Microsoft Query**.
3. Select **ServiceNow** as your database (the default DSN name).
4. Clear the **Use the Query Wizard to create/edit queries** check box.

**Note:** The Excel Query Wizard does not support the listing of columns from a table name that contain an underscore (_). Clearing this check box uses the Query Builder instead, which supports the use of this character.
5. Supply the ServiceNow user name and password that was pre-configured during the ODBC driver configuration procedure.

6. Select a table from the ServiceNow instance and click Add.

7. Close the dialog box.

8. Select the table columns from which the Query Builder will retrieve data. Use the list above the table, or type the names directly into the columns, and then press Enter.

9. To retrieve the data and create the Excel record, click the Return Data icon or select File > Return Data to Microsoft Office Excel.
<table>
<thead>
<tr>
<th>number</th>
<th>short_description</th>
<th>dv_assigned_to</th>
<th>dv_state</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC000009</td>
<td>Reset my password</td>
<td>David Loo</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000010</td>
<td>Need Oracle 10GF2 installed</td>
<td>Don Goodlife</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000011</td>
<td>Need new Blackberry setup</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000012</td>
<td>eFax is not working</td>
<td>David Loo</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000013</td>
<td>EMAIL is slow when an attach</td>
<td>Don Goodlife</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000014</td>
<td>missing my home directory</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000015</td>
<td>I can't launch my game anymore</td>
<td>Don Goodlife</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000016</td>
<td>Rain is leaking on main DNS S</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000017</td>
<td>How do I create a sub-folder</td>
<td>David Loo</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000018</td>
<td>Sales forecast spreadsheet is</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000019</td>
<td>Can't launch XWin32</td>
<td>Bud Richman</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000020</td>
<td>Request for a Blackberry</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000021</td>
<td>New employee hire</td>
<td>Beth Anglin</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000024</td>
<td>Issue with a web page</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000025</td>
<td>I need more memory</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000026</td>
<td>Seem to have an issue with my</td>
<td>Don Goodlife</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000028</td>
<td>My disk is still having issues C</td>
<td>Don Goodlife</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000027</td>
<td>please remove this host</td>
<td>ITIL User</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000029</td>
<td>I can't get my weather report</td>
<td>Don Goodlife</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000030</td>
<td>Lost connection to the wires</td>
<td>David Loo</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000031</td>
<td>EMAIL Server Down</td>
<td>David Loo</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000032</td>
<td>EMAIL Server Down Again</td>
<td>David Loo</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000033</td>
<td>File Server is 80% full - Needs</td>
<td>Don Goodlife</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000034</td>
<td>Does not look like a backup o</td>
<td>David Loo</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000035</td>
<td>Reset my password</td>
<td>Luke Wilson</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000036</td>
<td>Issue with networking</td>
<td>Luke Wilson</td>
<td>Open</td>
<td>1</td>
</tr>
<tr>
<td>INC000037</td>
<td>Request for a new service</td>
<td>Howard Inbaron</td>
<td>Open</td>
<td>1</td>
</tr>
</tbody>
</table>
The requested data is brought into Excel.
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INC00000009</td>
<td>Reset my password</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>2</td>
<td>INC00000010</td>
<td>Need Oracle 10GR2 installed</td>
<td>Don Goodlife</td>
<td>Open</td>
</tr>
<tr>
<td>3</td>
<td>INC00000011</td>
<td>Need new Blackberry setup</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>4</td>
<td>INC00000012</td>
<td>eFax is not working</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>5</td>
<td>INC00000013</td>
<td>EMAIL is slow when an attachment is involved</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>6</td>
<td>INC00000014</td>
<td>missing my home directory</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>7</td>
<td>INC00000015</td>
<td>I can’t launch my game anymore</td>
<td>Don Goodlife</td>
<td>Open</td>
</tr>
<tr>
<td>8</td>
<td>INC00000016</td>
<td>Rain is leaking on main DNS server</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>9</td>
<td>INC00000017</td>
<td>How do I create a sub-folder</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>10</td>
<td>INC00000018</td>
<td>Sales forecast spreadsheet is READ ONLY</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>11</td>
<td>INC00000019</td>
<td>Can’t launch X-Win32</td>
<td>Bud Richman</td>
<td>Open</td>
</tr>
<tr>
<td>12</td>
<td>INC00000020</td>
<td>Request for a Blackberry</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>13</td>
<td>INC00000021</td>
<td>New employee hire</td>
<td>Beth Anglin</td>
<td>Open</td>
</tr>
<tr>
<td>14</td>
<td>INC00000024</td>
<td>Issue with a web page</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>15</td>
<td>INC00000025</td>
<td>I need more memory</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>16</td>
<td>INC00000026</td>
<td>Seem to have an issue with my harddrive...</td>
<td>Don Goodlife</td>
<td>Open</td>
</tr>
<tr>
<td>17</td>
<td>INC00000028</td>
<td>My disk is still having issues. Can't delete file</td>
<td>Don Goodlife</td>
<td>Open</td>
</tr>
<tr>
<td>18</td>
<td>INC00000027</td>
<td>Please remove this hotfix</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>19</td>
<td>INC00000029</td>
<td>I can’t get my weather report</td>
<td>Don Goodlife</td>
<td>Open</td>
</tr>
<tr>
<td>20</td>
<td>INC00000030</td>
<td>Lost connection to the wireless network</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>21</td>
<td>INC00000031</td>
<td>EMAIL Server Down</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>22</td>
<td>INC00000032</td>
<td>EMAIL Server Down Again</td>
<td>David Loo</td>
<td>Open</td>
</tr>
</tbody>
</table>
Use the ODBC driver in Crystal Reports

After installing the ODBC driver and its associated DSN, use it in Crystal Reports as a data source provider.

**Note:** Crystal Reports includes the configuration file `CRConfig.xml` that contains the JVM minimum heap size (Xms) and maximum heap size (Xmx) values. When configuring the ODBC driver with Crystal Reports, ensure the ODBC driver uses the same minimum and maximum JVM heap size as Crystal Reports. If these values do not match, update the ODBC driver settings, not the Crystal Reports settings.

1. Create a new Standard Report

2. Create a new connection using the ServiceNow DSN
3. Select a table from the list of available tables
4. Select the available fields from the selected table
5. Click **Finish** to render the report
<table>
<thead>
<tr>
<th>Number</th>
<th>Short Description</th>
<th>Assigned To</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC00000008</td>
<td>Reset my password</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000009</td>
<td>Need Oracle 10gR3 installer</td>
<td>Don Goodenfield</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000011</td>
<td>Need new Blackberry setup</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000012</td>
<td>My email is not working</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000013</td>
<td>EMAIL is sent when an attachment</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000014</td>
<td>Missing my home directory</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000015</td>
<td>I can't launch my gnome screen</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000016</td>
<td>Rain is leaking on main DNS</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000017</td>
<td>How do I create a subdomain</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000018</td>
<td>Salesforce spreadsheet</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000019</td>
<td>Can't launch X-Win32</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000020</td>
<td>Request for a BlackBerry</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000021</td>
<td>New employee life</td>
<td>Beth Anglin</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000024</td>
<td>Issue with a webpage</td>
<td>ITIL User</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000025</td>
<td>I need more memory</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000026</td>
<td>System to have an issue with</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000027</td>
<td>My disk is still filling up</td>
<td>Don Goodenfield</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000028</td>
<td>My app is still failing</td>
<td>System Administrator</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000029</td>
<td>I can't get my weather report</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000030</td>
<td>Lost connection to the web</td>
<td>David Loo</td>
<td>Open</td>
</tr>
<tr>
<td>INC00000031</td>
<td>EMAIL Server Down</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>INC00000032</td>
<td>EMAIL Server Down Again</td>
<td>David Loo</td>
<td>Open</td>
</tr>
</tbody>
</table>
ODBC driver good practices

Follow these practices when using the ODBC driver to avoid common pitfalls and ensure optimal performance.

Avoid joins in queries

SQL joins used to combine tables negatively impact performance.

Use database views instead of joins when you need to query data from multiple tables. Database views perform essentially the same function as joins, while reducing the affect on system performance.

A number of useful database views are installed with the Database Views plugin, and administrators can create new database views.

Avoid views that reference themselves

When you use a view, ensure that it does not reference itself.

Calling a self-referencing view causes queries to loop and may result in a stack overflow error.

Avoid Top N statements

Statements that follow the format `SELECT TOP N`, such as `select top 10 number from incident;` impact instance performance.

| Note: This recommendation applies to the ODBC driver version 1.0.9 and earlier. Select Top N queries have improved performance starting with the 1.0.10 version. |

Instead of using this type of statement, construct queries that include a where clause so the ODBC driver pulls data in smaller batches based on indexed fields. For example, use `select number from incident where sys_updated_on = '<today's_date>';`.

Use `datadd` or `datediff` on literal values only

Avoid using the `datadd` and `datediff` statements when the result cannot be transformed to a literal value in the where clause.

Failing to return a literal value causes the driver to query all records from the table before filtering them, which may affect query performance.

Use the following example to determine when to use `datadd` or `datediff`. In the efficient query, the statement is transformed to `select number from incident where closed_at= '2015-05-05';`, allowing the driver to query fewer records from the instance.

<table>
<thead>
<tr>
<th>Table 1232: Example queries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inefficient query</strong></td>
</tr>
<tr>
<td><code>select number from incident</code></td>
</tr>
<tr>
<td><code>where datadd(day, curdate(), closed_at)=-1;</code></td>
</tr>
</tbody>
</table>

Reuse one session for multiple queries

When performing multiple queries, establish a connection to the instance before performing any queries and reuse this connection for all of the queries.

For example, if you use a `for` loop to iterate through multiple queries, establish the connection outside of the `for` loop, then perform each query within the `for` loop using the established connection.
Perform load testing

The ODBC driver uses SOAP message calls to execute queries. Use a load testing tool such as SoapUI to perform web services stress testing.

Be sure to include active user sessions during the load testing to better simulate production environment usage. If your SOAP load test performs poorly, contact support to request additional dedicated SOAP semaphores to manage your anticipated ODBC traffic.

Use an efficient WHERE clause

In queries with a WHERE clause that references a table column, always format the WHERE clause so that the column name appears before (to the left of) the condition.

For example, if you want to select all incidents that are active, the following is the correct way to write the query in order to improve performance:

```sql
select * from incidents
where active = 1
```

This same query using the following format may give invalid results or return an error:

```sql
select * from incident
where 1 = active
```

Use consistent timezones in queries

When you are performing time/date-bounded queries, ensure that both the client and users are set to the same timezone in order to avoid time lapses.

GlideRecord performs filtering based on the instance timezone and the ODBC client is filtered based on the Windows timezone. The result is an intersection of the two timezones which can cause a loss of queried information if these timezones do not match.

Workflows

Workflow provides a drag-and-drop interface for automating multi-step processes across the platform. Each workflow consists of a sequence of activities, such as generating records, notifying users of pending approvals, or running scripts. The graphical workflow editor represents workflows visually as a type of flowchart. It shows activities as boxes labelled with information about that activity and transitions from one activity to the next as lines connecting the boxes.

Explore
- Workflow release notes

Set up
- Getting started with workflow on page 3523

Administer
- Workflow roles on page 3642
- Administering workflow contexts on page 3643

Use
- Workflow editor on page 3525
- Create a workflow on page 3537
- Workflow activities on page 3676

Develop
- Developer training
- Developer documentation
- Workflow API reference
- Using variables in a workflow on page 3563

Upgrade
- Workflow movement with update sets on page 3652
- Workflow activity updates (pinning) on page 3650
Troubleshoot and get help

- Ask or answer questions in the Developer Community
- Troubleshoot workflows on page 3671
- Search the HI knowledge base for known error articles
- Contact ServiceNow Support

Getting started with workflow

The graphical workflow editor provides a drag-and-drop interface for automating multi-step processes across the platform.

Each workflow consists of a sequence of activities, such as generating records, notifying users of pending approvals, or running scripts.

The workflow starts when a triggering event occurs. Common triggers include a record being inserted into a specific table, or a particular field in a table being set to a specified value. For example, you might create a workflow that runs whenever a user requests approval for an item they want to order from the catalog.

When an activity completes, the workflow transitions to the next activity. An activity might have several different possible transitions to various activities, depending on the outcome of the activity. Continuing the example above, if the user's request is approved, the activity might transition to an activity that notifies someone to order the item; if the user's request is denied, the activity might transition to notifying the user that their request has been denied.

The graphical workflow editor represents workflows visually as a type of flowchart. It shows activities as boxes labelled with information about that activity and transitions from one activity to the next as lines connecting the boxes.

At each step in a workflow:

1. An activity is processed and an action defined by that activity occurs.
2. At the completion of an action by an activity, the workflow checks the activity's conditions.
3. For each matching condition, the workflow follows the transition to the next activity.

When the workflow runs out of activities, the workflow is complete.

Workflow activities

A workflow activity contains instructions that are processed by the workflow.

Activities can include running scripts, manipulating records, waiting for a set period of time, or logging an event. Workflow conditions determine whether or not the activity is performed. Activities can be added, removed, or rearranged. Transitions can be drawn between activities.

This is an activity that triggers a notification:
Transitions

After the workflow condition is evaluated, the workflow transition determines which activity is performed when the workflow condition is met.

This is a transition that always leads from the Change Approved script to the Change Task activity:

Exit conditions

After a workflow activity is performed, the workflow condition is evaluated to determine which transition is activated.

The condition determines behavior based on a change being approved or rejected:
Workflow editor

The Workflow Editor is an interface for creating and modifying workflows by arranging and connecting activities to drive processes.

You can manage multiple workflows in the same screen, create custom workflow activities, and use existing activities as data sources. Users with the workflow_creator role can create workflows. Users with the workflow_admin role can create, modify, delete, and publish workflows.

To open the Workflow Editor, navigate to Workflow > Workflow Editor. For information about using the editor, see Create a workflow on page 3537.

Workflow editing

The Workflow Editor is a user interface for defining workflows.

During workflow editing or while an unpublished workflow is running, only the person who checked out the workflow can view the changes.

After a workflow is published, it is available to other users. The workflow moves through the process as defined in the Workflow Editor. The entire workflow is represented in one screen. For example, this is the Standard Change workflow:
Figure 849: Standard change

Workflow interface

You interact with the workflow editor through several different elements of the window: the canvas, the canvas tabs, the title bar, the palette, and the palette tabs.

- **Canvas tabs**: Contains tabs for accessing workflows being edited or created.
- **Title bar**: Displays the workflow name and status. Provides a menu and controls for configuring, testing, and validating workflows.
- **Canvas**: Provides the working surface for creating new workflows or editing existing ones.
- **Palette tabs**: Contains tabs for accessing activities being edited or created.
- **Palette**: Contains all available workflow activities and existing workflows you can use as subflows. Drag activities and subflows to the canvas to create new workflows or edit existing ones.
Figure 850: Workflow user interface

Workflow canvas

The tabs on the workflow canvas allow a workflow designer to move between open workflows and access other resources.

The editor opens with the Welcome tab, which displays related help links and videos for basic workflows and Orchestration (when Orchestration is activated). From this tab, you can open articles in the ServiceNow Knowledge Base, user community conversations, live feed postings, and user documentation.

The drawing canvas is where you add activities and configure transitions for checked out workflows. Add an activity by dragging it from the palette to the workflow in the canvas. For more information, see Create a workflow on page 3537.
**Note:** If your organization blocks users from viewing YouTube videos, you can remove the video channel from either the workflow or Orchestration welcome pages using the procedure in *Remove a blocked video channel* on page 3534.
Workflow palette

The default workflow palette contains workflow activities and existing workflows you can use as subflows. The basic workflow palette contains these tabs:

- **Workflows**: Displays existing workflows and provides controls for creating new ones.
- **Core**: Displays baseline workflow activities available to all systems and Orchestration activities (when Orchestration is activated).

**Workflows tab**

The **Workflows** tab lists existing workflows that you can edit or use as subflows in other workflows. Double-click a workflow to open it in the canvas. To add a workflow as a subflow, drag it to another workflow in the canvas. Click the + icon to create a new workflow.

![Figure 852: Workflows tab in the palette](image)

**Core tab**

The **Core** tab contains the standard activities available by default to all workflows and any activities purchased with Orchestration, organized by category. Click the arrow icons to expand or collapse the activity lists under each category. To add an activity to a workflow, drag it to the canvas. For more information, see Add an activity to a workflow on page 3755.
Workflow editor title bar

When a workflow is opened in the canvas, the title bar displays the workflow title and the workflow status in italics. Possible states are Checked out by <name> and Published.
Controls on the right side of the title bar manage the workflow.

- **Workflow Properties**: Opens the current workflow's properties form.

- **Start**: Runs the workflow. This control is only available for workflows running on the Global table that are accessible from all application scopes. To test workflows that are on other tables, insert a record into that table that meets the condition of the workflow.

- **Validate**: Tests the workflow prior to publication. Validation detects potential problems that can prevent the workflow from publishing or cause the workflow to fail. For more information, see *Workflow Validation*.

- **Help**: Opens documentation to help you create the workflow.

**Workflow menu**

Click the menu icon in the title bar for additional options to configure the workflow.
These menu options are available:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Workflow</td>
<td>Creates a new workflow.</td>
</tr>
<tr>
<td>Open Existing</td>
<td>Opens another existing workflow.</td>
</tr>
<tr>
<td>Copy</td>
<td>Creates a duplicate of the workflow. Give the copy a different name.</td>
</tr>
<tr>
<td>Publish</td>
<td>Makes the personal workflow version public, overwriting the current published workflow version. This option is only available for checked out workflows.</td>
</tr>
<tr>
<td>Checkout</td>
<td>Creates a personal version of the workflow for you, which you can edit. This option is only available for published workflows.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the workflow. You cannot delete workflows that have contexts associated with them.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Set Inactive</td>
<td>Inactivates the workflow so that it cannot be used.</td>
</tr>
<tr>
<td>Expand Transitions</td>
<td>Redraws the transitions so that they do not overlap when they leave the activity condition.</td>
</tr>
<tr>
<td>Start Workflow</td>
<td>Starts a test run of the current workflow.</td>
</tr>
<tr>
<td>Validate Workflow</td>
<td>Runs validation tests on your workflow prior to publication. Use this validation to detect potential problems that can prevent the workflow from publishing or cause the workflow to fail. For more information, see Validate a workflow from the workflow editor on page 3549.</td>
</tr>
<tr>
<td>Collapse Transitions</td>
<td>Redraws the transitions so they overlap when they leave the activity condition.</td>
</tr>
<tr>
<td>Properties</td>
<td>Opens the Workflow Properties form, which defines the workflow's attributes.</td>
</tr>
<tr>
<td>Edit Inputs</td>
<td>Opens the Workflow Inputs list of variables that the workflow can accept when used as a subflow. For more information, see Invoke a subflow in a workflow on page 3557.</td>
</tr>
<tr>
<td>Edit Stages</td>
<td>Opens the Workflow Stages list. For more information, see Workflow stages on page 3587. For tables with a column of Type = Workflow.</td>
</tr>
</tbody>
</table>

Remove a blocked video channel

If your organization blocks users from downloading the videos on the Workflow and Orchestration welcome pages, you can remove the videos.

The instructional videos on the workflow and Orchestration welcome pages are hosted on YouTube. Some organizations block viewing videos from this site. You can follow this procedure to remove the videos from either welcome page.

1. Navigate to System UI > UI Pages.
2. Open the workflow_editor_welcome record.
3. From the HTML field, remove the lines shown here for one or both of the welcome pages and update the record.

   - Basic workflow

   ```html
   <div id = "workflowchannel" class="widget-cotent">
      <h3 class="widget-header"> ${gs.getMessage('Channel')} </h3>
   
   <div class="widget-link">
   </div>
   
   <iframe src="https://www.youtube.com/embed?list=PLCoMiTb5WX3o_ksSvCnhk1AfAh4yZOZ1Yz" width="525" height="380" frameborder="0" allowfullscreen="true"></iframe>
   </div>
   ```
- **Orchestration**

```html
<div id="orcheschannel" class="widget-content">
  <h3 class="widget-header"> ${gs.getMessage('Channel')}</h3>
  <div class="widget-link">
  </div>
  <iframe src="https://www.youtube.com/embed?list=PLCOmiTb5WX3o6xH1f0Z0Xm" width="525" height="380" frameborder="0" allowfullscreen="true"></iframe>
</div>
```
UI Page - workflow_editor_welcome

Name: workflow_editor_welcome
Category: General
Application: Global
Description: Welcome page for workflow editor

```html
</li>
</ul>
</div>
</div>

<div class="content-body">
  <div class="widget-container">
    <h3 class="widget-header">${gi.getMessage('Channel')}</h3>
    <div class="widget-link">
    </div>
    <iframe src="http://www.youtube.com/embed/8CZHsbSkX1k?rel=0" width="525" height="390" frameborder="0" allowfullscreen="true"></iframe>
  </div>
</div>

<div class="widget-container">
  <h3 class="widget-header">${gi.getMessage('Community')}</h3>
  <div class="widget-link">
    <a target="_new" href="https://community.servicenow.com/discussions/createsupportinput.jsp?al=containerType=GMO&containerId=2015">Start a Dialog</a>
  </div>
</div>

<iframe src="${serviceNowURL}/render_gadget_widgetWorkflowRSS.do?sysparm_nostack=true" width="525" height="390" frameborder="0" allowfullscreen="true" scrolling="no"></iframe>
</div>
</div>
</body>
</html>
```
Create a workflow

Automate a multi-step process by creating a workflow with the workflow editor.

- You must have the workflow_admin or workflow_creator role to use the workflow editor.
- If you are designing the workflow as part of an update set process, see Workflow movement with update sets on page 3652 before creating the workflow.

1. Navigate to Workflow > Workflow Editor.
   The Welcome tab of the Workflow Editor opens. It links to workflow documentation and other related resources. If ServiceNow Orchestration is activated, the welcome screen contains resources for that feature as well.

2. On the Workflows tab in the palette, click the + icon in the upper right corner.
   A simplified version of the New Workflow form opens.

3. Fill in the Name and Table fields
4. [Optional] Add a Description.

5. [Optional] Do one of the following:
   a) If the Conditions UI section is displayed, specify a Condition if needed and edit the fields.
      (The Conditions UI section shows only if the selected table supports conditions for launching workflows. For example, if you select the sc_req_item table, conditions are not applicable and the Conditions UI section is not displayed.)
   b) If the Stages UI section is displayed, check that the State rendering and Stage order fields contain the correct information. (The Stages UI section shows only if the selected table supports stages. For example, if you select the sc_req_item table, the Stages UI section is displayed.)

6. Click Submit.

   The new workflow is created with the Begin and End activities connected by a single transition.

7. Finish creating the workflow by adding activities, validating, and publishing so the workflow is available to other users.
   For more information, see Add a workflow activity on page 3549, Validate a workflow from the workflow editor on page 3549, and Publish a workflow on page 3551.

8. To change advanced settings for the workflow, click the Properties icon. If you make changes, click Update.

Create a workflow from a table

Automate a multi-step process by creating a workflow from the list view of any table that supports workflows.

- If you are designing the workflow as part of an update set process, see Workflow movement with update sets on page 3652 before creating the workflow.

1. Open a table, such as Incident or Problem, in list view. For example, navigate to Incident > Open.
2. Right-click in the column header and select Configure > Workflows.
The Workflow Versions on that table appear in a list.

3. Click New.
   The Workflow Version opens in New Workflow view. The Table field is filled in with the table you selected in step 1 and is read-only.

4. Enter Name.

5. [Optional] Enter Description.

6. [Optional] Edit conditions fields as necessary.

7. Click Submit.
   The new workflow is added to the Workflow Versions list.

8. Click the workflow Name.
   The new workflow is created with the Begin and End activities connected by a single transition.
9. Finish creating the workflow by adding activities, validating, and publishing so the workflow is available to other users.
   For more information, see *Add a workflow activity* on page 3549, *Validate a workflow from the workflow editor* on page 3549, and *Publish a workflow* on page 3551.

10. [Optional] To change advanced settings for the workflow, click the Properties icon. If you make changes, click Update.

### Create a workflow for a new service catalog item

When you create a new service catalog item, you can create a new corresponding workflow at the same time.

- If you are designing the workflow as part of an update set process, see *Workflow movement with update sets* on page 3652 before creating the workflow.

1. Navigate to **Service Catalog** > **Catalog Definitions** > **Maintain Items**.
2. At the top of the form, next to **Catalog Items**, click **New**.

   The Catalog Item form opens.

3. Add a **Name**.
4. Next to the **Workflow** field, click the lookup icon.
5. Next to Workflow at the top, click **New**.
The Workflow version dialog opens in the New Workflow View. The **Table** field is set to **Requested Item (sc_req_item)** and is read-only.

6. Add a **Name**.

7. [Optional] Add a **Description**.

8. [Optional] Change the stage information as necessary.

9. Click **Submit**.
The new workflow is created with the **Begin** and **End** activities connected by a single transition.

10. Finish creating the workflow by adding activities, validating, and publishing so the workflow is available to other users.
    For more information, see *Add a workflow activity* on page 3549, *Validate a workflow from the workflow editor* on page 3549, and *Publish a workflow* on page 3551.

11. To change advanced settings for the workflow, click the **Properties** icon.

12. Click **Update**.
    If you close the workflow editor, you can see the Catalog Item record. Note that the workflow is added to the Workflow field. The Show Workflow and Information icons appear next to the **Workflow** field. Hover over the information icon to view a read-only summary of the workflow.

### Create a workflow for an SLA Definition

Automate a multi-step process by creating a workflow from an SLA definition.

- If you are designing the workflow as part of an update set process, see *Workflow movement with update sets* on page 3652 before creating the workflow.

1. Open a list of SLA definitions. For example *Facilities > SLA Definitions* or *Service Level Management > SLA Definitions*.

2. At the top of the form, next to **SLA Definitions**, click **New**.

3. Select **Contract SLA** or **Service Offering SLA**.
   The SLA Definition or Service Offering SLA form opens.

4. Next to the **Workflow** field, click the lookup icon.

5. Next to **Workflow** at the top, click **New**.
The Workflow Version dialog shows in the New Workflow view. The Table field is set to SLA Definition (contract_sla) or Service Offering SLA (service_offering_sla) and is read-only.

6. Enter Name.
7. [Optional] Enter Description
8. [Optional] Edit conditions fields as necessary.
9. Click Submit.
   The new workflow is created with the Begin and End activities connected by a single transition.
10. Finish creating the workflow by adding activities, validating, and publishing so the workflow is available to other users.

For more information, see Add a workflow activity on page 3549, Validate a workflow from the workflow editor on page 3549, and Publish a workflow on page 3551.

11. [Optional] To change advanced settings for the workflow, click the Properties icon. If you make changes, click Update.

Workflow properties

In the properties of a workflow, you can configure settings such as its application scope, start conditions, schedule, inputs, stages, and run time metrics. You can also view information such as the workflow author, version, and history.

When you create a new workflow, the following fields are available in the dialog box:

- Name
- Table
- Description
- If condition matches
- Condition

If you click Diagrammer view in Related Lists, the following UI sections are available in the dialog box:

- General
- Conditions
- Inputs
- Activities
- Application
- Schedule
- Stages
- Estimated Runtime
## General

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name to identify the workflow.</td>
</tr>
<tr>
<td>Table</td>
<td>The table for the workflow to run on. Workflows that run on specific tables can still interact with other tables. Select <strong>Global</strong> to run the workflow on all tables.</td>
</tr>
</tbody>
</table>

**Note:** The list shows only tables and database views that are in the same scope as the workflow. Also, all users who edit the workflow must have access to the necessary tables and domains.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checked out</td>
<td>[Read-only] When the workflow was checked out. Automatically set by the <strong>Checkout</strong> action in the workflow menu.</td>
</tr>
<tr>
<td>Checked out by</td>
<td>[Read-only] The user who has this workflow checked out. This value is automatically set by the <strong>Checkout</strong> action in the workflow menu.</td>
</tr>
<tr>
<td>Published</td>
<td>[Read-only] Check box to indicate whether the workflow has been published. Automatically set by the <strong>Publish</strong> action in the workflow menu.</td>
</tr>
<tr>
<td>Description</td>
<td>The purpose of the workflow.</td>
</tr>
</tbody>
</table>

## Conditions

Create conditions to trigger the workflow. The Conditions section does not appear if you select a table, such as sc_req_item, that does not require a condition.
If condition matches

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When the condition evaluates to true, the workflow launches an active context:</td>
</tr>
<tr>
<td>• <strong>None</strong>: The workflow is not automatically started by the workflow engine. To run this workflow, write a script to start the workflow.</td>
</tr>
<tr>
<td>• <strong>Run the Workflow</strong>: The default value. The workflow engine starts the workflow if the information in the <strong>Condition</strong> field matches a record that is inserting into the table.</td>
</tr>
<tr>
<td>• <strong>Run if no other workflows match yet</strong>: The workflow only runs if no other workflows are running on a specific record. For example, there are four workflows inserted into the Incident table, which have a condition such as <code>short_desc</code> contains <code>test</code>. A new workflow, which has <strong>If condition matches</strong> is set to <strong>Run if no other workflows match yet</strong>, only runs if none of the four workflows have started running on the Incident record.</td>
</tr>
</tbody>
</table>

**Condition**

A condition builder for specifying workflow conditions that trigger the behavior selected from the **If condition matches** list.

**Order**

Numeric value that determines the order of the workflow, relative to other workflows. Workflows are evaluated in order from the lowest order number to the highest. A workflow runs if it is the first to match conditions.

---

**Inputs**

The Inputs section lists all the activities in the current workflow that input data, the data type, and the default value. The Inputs section is only available after a workflow has been created. To create a variable, click **New**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Displayed column label. Localized depending on user locale.</td>
</tr>
<tr>
<td>Reference</td>
<td>Input field from another table.</td>
</tr>
<tr>
<td>Type</td>
<td>Data type. For example, integer or string.</td>
</tr>
<tr>
<td>Default value</td>
<td>Value used if you do not provide a value.</td>
</tr>
</tbody>
</table>

**Activities**

The Activities section enables you to set activity pinning and maximum activity count.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Activity pinning             | List of options that control updates to custom activities at the workflow level. Pinning protects custom activities from being updated automatically when downloaded from the ServiceNow Store. For more information, see Workflow activity updates (pinning) on page 3650. The possible options are:  
  - **Set by activity**: Allows all activities in the workflow to use their own pinning settings. This is the default pinning option.  
  - **Pin all activities**: Pins all activities in the workflow to their current version.  
  - **Unpin all activities**: Allows all activities in the workflow to be updated. |
<p>| Max activity count           | The maximum number of activities performed by the workflow. This value is used to prevent infinite loops and is set to 100 by default. When the stated maximum count is reached, the workflow is canceled. If this field is blank, the maximum count is set to -1, and the workflow is canceled. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery based on</td>
<td>The schedule type for this workflow. Possible types are:</td>
</tr>
</tbody>
</table>
|                       | • **User-specified duration**: Duration based on a user-specified value. This is the default schedule type.  
|                       | • **Relative duration**: Duration calculated from a preconfigured schedule, such as 8-5 weekdays. |
| Expected time         | User-defined interval. This field is visible when the schedule type is **User-specified duration**. |
| Schedule              | Preconfigured schedule that determines when this workflow runs.             |
| Timezone              | Time zone for this instance.                                               |

### Stages

The Stages section appears if you select a table with **Type = Workflow**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage</td>
<td>Displays the workflow stage progress on the selected table. Optionally, select <strong>Stage rendering</strong> and <strong>Stage orders</strong> schemes to customize the appearance of the stage field. The default values cover typical scenarios.</td>
</tr>
<tr>
<td>Stage rendering</td>
<td>The renderer to use when displaying stage icons on a form or list view. For more information about renderers, see <strong>Workflow stage renderers</strong> on page 3628.</td>
</tr>
<tr>
<td>Stage order</td>
<td>The order of workflow stages when you view a workflow field in a list. Select <strong>Computed</strong> to let the workflow engine compute the stage order from the order of execution in the workflow. Select <strong>User Specified</strong> to use the <strong>Order</strong> field from that <strong>Workflow stages</strong> on page 3587.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the stage as it appears in workflow fields..</td>
</tr>
<tr>
<td>Duration</td>
<td>Time allocated for the specific stage.</td>
</tr>
<tr>
<td>Order</td>
<td>The order of workflow stages when you view a workflow field in a list. Select <strong>Computed</strong> to let the workflow engine compute the stage order from the order of execution in the workflow. Select <strong>User Specified</strong> to use the <strong>Order</strong> field from that <strong>Workflow stages</strong> on page 3587.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Value</td>
<td>The value of the stage when it is referenced from elsewhere in the system, such as in a script.</td>
</tr>
</tbody>
</table>

### Estimated Runtime

The **Estimated Runtime** section opens the controls for configuring the ERT for the workflow. Core workflows included in the base system are not configured for estimated run time by default. All new workflows are configured with default ERT values automatically. You can edit existing run time estimates or configure new ones for any existing workflow. For details about how estimated run times are configured and calculated, see *Workflow run time metrics* on page 3670.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires ERT</td>
<td>Check box to indicate that this workflow requires an estimated runtime configuration.</td>
</tr>
<tr>
<td></td>
<td>You can use the ERT calculations to determine if workflows are running longer or shorter than expected and to identify errors in workflow processing. By default, new workflows require an ERT.</td>
</tr>
<tr>
<td>Estimated Run Time</td>
<td>The initial estimate for the workflow’s run time.</td>
</tr>
<tr>
<td>Number of data points</td>
<td>[Read only] The number of times the system has compared the estimated run time to an actual run time result.</td>
</tr>
<tr>
<td>Outlier Percentage Threshold for ERT</td>
<td>[Required] The percentage deviation from the estimated run time that identifies an outlier workflow run time. The system uses a default value of 20. For more information, see <em>Outlying workflow run times</em> on page 3670.</td>
</tr>
</tbody>
</table>

### Add a workflow activity

Activities determine the functionality of the workflow.

When they are created, all workflows contain **Start** and **End** activities.

For more information, see *Workflow activities* on page 3676.

1. Open a workflow.
2. Check out the workflow.
3. **Drag a workflow activity** from the Activities menu into the workflow body.
4. Populate the Workflow Activity form that appears.

### Validate a workflow from the workflow editor

You can manually validate a workflow from the workflow editor.

Role required: workflow_admin
Running a workflow on a new node automatically attempts to validate the workflow. If validation is successful, the system updates the workflow version record to indicate the workflow has been validated and marks the record as updated by the user who ran the workflow.

1. Open the workflow to validate in the workflow editor.
   When the workflow is loaded, the workflow validator icon appears in the toolbar.

2. Click the validator icon to run a series of validation tests on the current workflow version and generate a report.

<table>
<thead>
<tr>
<th>Type</th>
<th>Level</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValidateLowestCommonTable</td>
<td>info</td>
<td>The lowest common table in this workflow is.</td>
</tr>
<tr>
<td>ValidateTransitionIn</td>
<td>info</td>
<td>All activities in this workflow have at least one inbound transition.</td>
</tr>
<tr>
<td>ValidateInputVarUpdateSetDependencies</td>
<td>info</td>
<td>There were no input Variable Update Set dependency issues found.</td>
</tr>
<tr>
<td>ValidateDanglingTransition</td>
<td>info</td>
<td>There are no unattached transitions in this workflow.</td>
</tr>
<tr>
<td>ValidateParentFlow</td>
<td>warn</td>
<td>This workflow version ([Item Designer - Approvals] is required as a subflow in 1 other workflows.</td>
</tr>
<tr>
<td>ValidateScriptForCurrentDotUpdate</td>
<td>info</td>
<td>The JavaScript in this workflow has no instances of 'currentDotUpdate'</td>
</tr>
<tr>
<td>ValidateWorkflowHasColumn</td>
<td>info</td>
<td>Workflow columns are valid.</td>
</tr>
</tbody>
</table>
Publish a workflow

When a workflow is complete, publish the workflow so that it is available to all users.

Before you publish a workflow, validate it to test it for issues that might cause it to fail, such as missing subflows or disconnected transitions. For more information, see Workflow validation on page 3589.

To publish a workflow:

1. Navigate to Workflow > Workflow Editor.
2. Open the workflow that you want to publish.
3. In the title bar, click the menu icon and select Publish.

If you published a new version of workflow, the changes are not applied to running workflow contexts. Any currently running workflow context continues using the workflow version that was available when the workflow started. The next time the workflow runs, it uses the new version.

Ending workflows with multiple branches

A workflow is complete when it reaches the End activity, even if there are still active branches of the workflow in progress. To ensure that both branches are completed, add a Join activity to resolve the branches.

For example, the following figure shows a workflow with two branches that execute independently. When Task 1 and Task 2 of Branch B are completed, the workflow is marked complete even if the Branch A tasks are not completed.

![Figure 856: Workflow with uncompleted branch](image)

For both branches to complete, add a Join activity to resolve the branches. When one branch reaches the join, the workflow waits for the other branch. When both branches are complete, the workflow reaches the end. The Incomplete condition of the a Join activity is met only if one of the branches cannot be completed.
Edit a published workflow

You can edit a published workflow after you check it out.

**Note:** You cannot check out or delete workflows that are associated with a read-only application file.

To check out a workflow:

1. Navigate to **Workflow > Workflow Editor**.
2. Open the workflow that you want to edit.
3. In the title bar, click the menu icon and select **Checkout**.
   
   A new version of the workflow is created and assigned to you.
   
   If you are in a different domain than the published workflow, the new workflow version is created in your domain.

After you finish editing the workflow, validate and publish the workflow to make the new version available to other users.

1. **Validate a workflow from the workflow editor** on page 3549
2. **Publish a workflow** on page 3551

Workflows used as subflows

A workflow can launch another workflow as an activity.

The parent workflow triggers the subflow and then waits for the subflow to complete before continuing. Run the workflow validation tool prior to publishing to detect missing subflows and other dependency problems, such as those involving update sets.

The **Workflows** tab in the Workflow Editor contains a list of the workflows available for use as subflows.
Figure 858: Workflows available to use as subflows

Make sure that the selected subflow is active. If the subflow is inactive, the main workflow will hang with a Loading message. If you place an inactive subflow into a workflow, the subflow appears with a red banner, indicating that it cannot run. An active subflow is highlighted in blue when selected.
Subflows and the Create Task activity

If a workflow contains a Create Task activity that has executed on the current record, additional task activities in the workflow might not execute as expected.

This can happen when the same subflow containing a Create Task activity runs more than once in a parent flow. When the subflow reruns and attempts to execute the Create Task activity again, the system reopens the first task activity instead and does not create an additional task.

**Note:** An alternative to creating duplicate subflows that use the Create Task activity is to add a Run Script activity to the workflow that creates a task with a script.
Figure 860: The same create task activity runs twice in a workflow

In this configuration, the workflow does not run the same subflow containing a Create Task activity more than once. This allows the workflow to create additional tasks.
Pass a variable from a workflow to a subflow

Use this process to pass variables from a parent workflow to a subflow.

**Note:** You can also use the *Return Value* activity in the subflow to return values to the parent workflow. Make sure to have a *Return Value* on every ending transition path.

1. Prepare the subflow to accept variables from the parent workflow by defining the inputs.
2. Include the subflow in the parent workflow and connect the inputs to the parent workflow variables.

**Note:** Subflows on the Requested Item [sc_req_item] table in the Service Catalog cannot accept variables from parent workflows.

**Define inputs for a subflow**

Define the input variables for a workflow to request from parent workflows when it is launched as a subflow.

All inputs are stored in the Variables [var_dictionary] table.

1. In the editor, open and check out the workflow.
2. In the title bar, click the menu icon and select Edit Inputs.
3. In the Workflow Inputs window, click New.
4. Populate the record with the definition of the variable, including the column name, the label that is displayed to the user, and the type of field.
5. Click **Submit**.

*Invoke a subflow in a workflow*

Use this procedure to add a subflow to a workflow.

1. In the Workflow Editor, open and check out the parent workflow.
2. Drag the subflow from the **Workflows** tab to the parent workflow.
3. In the **New Activity** dialog box, define the variables defined by the subflow's **Inputs**.
   These fields can accept both static values or variables in the following format:
   
   \( ${\text{variable_name}} \)

4. Click **Submit**.
   The workflow triggers the subflow at the appropriate time and passes the variables as indicated by the **Inputs** definition.

*Prepare a subflow*

This example describes the process of preparing a subflow for use in a parent workflow.

1. In the editor, open and check out the workflow that you want to use as a subflow.
2. In the title bar, click the menu icon and select **Edit Inputs**.

   ![Edit Inputs Menu]

3. In the Workflow Inputs window, click **New** in the **Variables** list.
4. Add a new variable depending on the type of values that it is going to store. The following example sets up a string value.

5. Click **Submit**.

6. Close the **Workflows Inputs** dialog.

7. Create a **Run Script** activity on the subflow.
   - Set the value from the parameter to a field on the current form. This is important because the **Notification** activity can only pull values from the current variable and not from the newly added variable. The following example sets the value in the **Description** field.
     
     ```
     current.description = workflow.inputs.bluesubvariable;
     ```
   - Create a new field on the request form but do not display the field. This serves as temporary storage.
8. Create a **Notification** activity on the subflow. Use `$description` in the subject to return the value from the field.
This is what the subflow would look like:
Prepare a workflow to use a subflow

After you create a subflow, use this procedure to prepare the parent workflow.

1. On the parent workflow, create a variable similar to what you did on the subflow, but name it something different.
   In the following example, the variable is named **Blue Main Variable**.

   ![Variable creation interface]

2. Click **Submit**.

3. Insert a **Run Script** activity to return the value from a field to the newly created variable.
   In this example, the value of the **Short Description** field is returned and given to the newly created variable.

   ```script
   workflow.scratchpad.bluemainvariable = current.short_description;
   ```
4. Click **Submit**.

5. In the subflow activity, set the **Blue Sub Variable** to pass the `bluemainvariable` to the `bluesubvariable`.

   ```plaintext
   workflow.scratchpad.bluemainvariable = current.short_description;
   ```

   This is what the main workflow looks like:
Using variables in a workflow

Within workflow there are several different types of variables that are available.

Some variables define, describe, or compose the pieces of the workflow (such as workflow activities). Some variables are available specifically to facilitate the execution of a workflow (such as scratchpad variables). Some variables define, describe, or compose the data records being manipulated by the workflow process (for example, the elements of a Glide Record that are defined in a dictionary.xml file describing a record, such as an incident).

Activity variables

Activity variables are properties specifically associated with a workflow activity.

These are the inputs that appear on the activity definition when a specific activity is dragged onto the workflow canvas. These variables are properties that define, compose or describe the workflow activity, or the work it is designed to perform.

Activity variables are declared in a related list within the Workflow Activity Definition.

To add, view, or modify a variable:

1. Navigate to Workflow > Activity Definitions.
2. Select a workflow activity, such as Approval - User.
3. In the Activity Variables section or tab, add, view, or modify the variables. Activity variables are defined using the same fields as Dictionary records.

Note: If defining a new activity variable that is dependent on another activity variable, put the prefix “vars.” before the name of the parent activity variable. For example, if the child field is dependent on a field named parent, put the value vars.parent in the Dependent field.

To control the visibility of an activity variable on the workflow canvas after the activity has been dragged to the canvas, use a UI policy on the wf_activity table.

To access the variables or assign values to these variables within the workflow activity, use the vars variable of the workflow activity.

activity.vars.variable_name = new_variable_value

Workflow input variables

Workflow variables are external values that are passed into and referenced by a workflow during its execution.

Workflow variables can be declared externally in a script and passed into a main workflow or they can be declared within a main workflow and passed as inputs to a subflow.

Note: There is a specific kind of input variable for any workflow that is declared against the table Requested Item [sc_request_item]. To understand the use of these variables see Workflow catalog variables on page 3565.

Workflow variables are declared in a form available from the Workflow Editor gear menu.

To add, view, or modify a variable:

1. Navigate to Workflow > Workflow Editor.
2. Edit and checkout a workflow.
3. Select the Workflow Activities menu and select Edit Inputs.
4. Select an existing variable or click New.
Note the **Column name** field. Use this value when accessing the variable from a script.

Workflow variables do not appear on the workflow canvas, they are only available for view from the Workflow Activity menu.

The input variables are available to use in scripts. To access an input variable from a script:

```javascript
var input  = workflow. variables. variableName //get the workflow variable identified by column name.
```

### Assigning values to variables from outside of workflow

After being declared in a workflow, values can be assigned to the variables from where the workflow is kicked off.

The following example demonstrates how a workflow variable's value can be set in a script include, business rule, or UI action:

```javascript
//Declare an instance of workflow.js
var wf  = new Workflow ( ) //Get the workflow id
var  wfId  = wf. getWorkflowFromName ( "Workflow Name" ) ;
//Start workflow, passing along name : value pair(s) for mapping to variable
//where input_var_name is the name of the variable declared in gear menu
//and input_var_value is whatever that value should be for this execution of workflow
wf. startFlow (wfId , null , "Workflow Name" , {input_var_name : input_var_value } ) ;
```

Another example that demonstrates a more readable way of passing multiple variables involves declaring an object, adding the names/values to the object, and passing it:

```javascript
//Declare an instance of workflow.js
var wf  = new Workflow ( )
//Get the workflow id
var  wfId  = wf. getWorkflowFromName ( "Workflow Name" ) ;
//Start workflow, passing along object containing name/value pairs mapping to inputs expected by the workflow
var vars  = { } ;
vars. input_var_name1 = input_var_value1 ;
vars. input_var_name2 = input_var_value2 ;
// add as many variables as your workflow is expecting, then pass the object
wf. startFlow (wfId , null , "Workflow Name" , vars ) ;
```

### Assigning Values for Subflow Inputs from inside of a Main Workflow

If a workflow that accepts inputs is called from within another workflow, those input values can be set from the workflow canvas within the UI of the workflow activity. Each input variable declared for a subflow will have a text box input area displayed on the activity. Within these text boxes, the variables can be set with any value either from the scratchpad or from within the current record.

For more information, see **Workflows used as subflows**.
Reading the Value of a Workflow Input Variable

The value of a Workflow input variable is accessible anywhere within the workflow that accepts javascript by dot walking the current workflow object:

```javascript
var readValue = workflow.inputs.variable_name;
```

Workflow catalog variables

Workflows that are associated with the Requested Item [sc_req_item] table have a specific type of workflow input.

The inputs into this workflow are essentially question:answer pairings that, when associated with a specific catalog task, become options on the task form. These options are generated by that catalog task activity within a workflow.

Scope

For the purposes of Service Catalog and workflow, a variable’s scope determines its availability to a catalog task activity within a workflow. You can define variables as global or catalog item-specific. When a variable is associated to a catalog item and the workflow generating the catalog task is associated to the catalog item, the variable scope determines which variables are available for mapping within the catalog task activity in a workflow. If a workflow is associated with Requested Item [sc_request_item], and is not specifically associated with a catalog item, any catalog variable with a blank Cat Item field value is available to all catalog task activities within the workflow.

Declaration

You can declare requested item catalog variables at the catalog item level or at a global level.

You can declare global catalog variables using the workflow editor context menu.

Also see, Global variable declaration option 2.

1. Navigate to Workflow > Workflow Editor.
2. Open and check out a workflow that runs against the Requested Item [sc_req_item] table.
3. Click the menu icon and select Edit Catalog Variables.
4. Click New to create a new variable or select an existing variable.
5. Select the Global check box.
6. Click Submit.

Global variable declaration option 2

You can declare global catalog variables using the Service Catalog.

1. Navigate to Service Catalog > Item Variables.
2. The list of variables that appears in the workspace has a default filter of Cat Item != <blank>. Change the filter to Cat Item = <blank> and run the query.
   The catalog variables that appear are available to any catalog task that is initiated from within a workflow.
3. Click New to create a new variable.
4. Leave the Cat Item field blank.
5. Select the Global field blank.
6. Click Submit.

Also see, Global variable declaration option 1.

© 2017 ServiceNow. All rights reserved. 3565
Catalog item specific variable declaration option 1
You can declare a catalog item-specific variable.

These variables are only available to the catalog item referenced in the Cat Item field on the variable record.

1. Navigate to Service Catalog > Catalog Variables > Item Variables.
   Note the list of variables that appears in the workspace has a default filter of Cat Item != <blank>.
2. Enter or select a catalog item in the Cat Item reference field.
   Selecting a Cat Item restricts the scope and availability of the catalog variable to that specific catalog item.
3. Click Submit.
   Also see, Catalog item specific variable declaration option 2.

Catalog item specific variable declaration option 2
You can declare a catalog item-specific variable directly from a catalog item record.

Creating a catalog variable in this way automatically sets the Cat Item reference to the catalog item selected.

1. Navigate to Service Catalog > Catalog Definitions > Maintain Items.
2. Select or create an a catalog item.
   Note the Variables related list. All variables declared using this related list have a Cat Item reference value of the current catalog item.
3. From the Variables related list, click New.
4. Ensure that the Cat Item field correctly references the catalog item previously selected or created.
5. Go to or open the Question section or tab.
6. Add a Question.
7. Add a Name.
   Variable names should not include white space and cannot begin with a number.
8. Click Submit.
   Also see, Catalog item specific variable declaration option 1.

Display
Catalog specific item variables are visible in several places depending on where in the process the variable is viewed.

Declaration can happen and variables can be seen from within the menu, inside both the Maintain Items module and in the Item Variable modules of the Service Catalog.

Within a workflow, the Catalog Specific Item variables are available to the Catalog Task activity in the form of a slushbucket at the bottom of the Catalog Task activity. Item variables that are selected, are the question and answer pairs that will appear on the task that is generated by that instance of Catalog Task when executing that workflow.

If a workflow is associated with a specific catalog item, the association acts as a filter for item variables that appear in the slushbucket of the Catalog Task activity.

The last place the variables are seen is in the task form that is generated by the Catalog Task Item. The variables selected in the slushbucket are the question and answer pairs that appear to the user on the task form.

Access and assignment
The Catalog Item Variables are available and assigned to a specific Catalog Task activity (thereby to a specific task) using the slushbucket entry of a catalog task.

The user working the task enters the values of the variables.
To access the values of a Catalog Item Variable inside a script:

```javascript
var gr = current; // or create and query a new GlideRecord
var itemVariable = gr.variables[variableName]; // access the service catalog variable identified by the variable name.
var itemValue = itemVariable.getGlideObject().getValue();
var itemQuestion = itemVariable.getGlideObject().getQuestion().getLabel();
```

**Workflow scratchpad**

The *scratchpad* is a special field on a workflow context that allows workflow activities to communicate data to subsequent activities.

The scratchpad is available within the workflow as a variable (*scratchpad*), and variables in the scratchpad are available as properties on that variable. Using the scratchpad requires at least two activities in a workflow: the sending activity, which writes data to the scratchpad, and the follow-up activity, which uses this data.

The scratchpad can hold variables of any JavaScript data type that can be represented as JSON. These data types include primitives such as *String* or *Boolean*, or JavaScript objects such as *Object*, *Array*, and *Date*. You cannot add functions or scriptable Java objects, such as *GlideRecord*, to the scratchpad. For an alternative method of using the scratchpad with scriptable Java objects, see *Scriptable Java objects* on page 3569.

The properties of supported JavaScript objects can be objects or arrays themselves. For a description of the data types supported in a scratchpad variable, see *Introducing JSON*.

**Workflow scratchpad variables**

The scratchpad is a space in the workflow context to store and share string based variables (as *name:value* pairs) between instances of activities within an executing instance of a workflow.

The scratchpad is global to the instance of the running workflow and as such, is available equally to all activities.

**Declaration**

The scratchpad itself is automatically available to an executing workflow and requires no specific declaration. Variables are declared and stored in the scratchpad simultaneously by referencing it. For example:

```javascript
workflow.scratchpad.variableName = variableValue;
```

**Display**

There is no way to view the contents of the scratchpad.

**Access and Assignment**

As stated in the Declaration section:

```javascript
workflow.scratchpad.variableName = variableValue;
```
or:

```javascript
var myValue = workflow.scratchpad.variableName;
```

**Store multiple data types in the workflow scratchpad**

The workflow scratchpad can store multiple data types.

Data types the scratchpad can store include:
- Primitives such as integers, boolean values, or strings.
- JavaScript objects such as user data.
- Scriptable Java objects such as GlideRecord or GlideDateTime.

Add primitives to the workflow scratchpad

You can add any primitive value to the scratchpad.

```javascript
// numbers
workflow.scratchpad.maxdays = 100;
workflow.scratchpad.pi = 3.14159267;

// booleans
workflow.scratchpad.canSave = true;
workflow.scratchpad.isHoliday = isHoliday( new Date() );

// string
workflow.scratchpad.next_holiday = 'Christmas';
```

Add JavaScript objects to the workflow scratchpad

You can add any JavaScript object to the scratchpad if the object can be represented as JSON.

Whenever possible, group related values into a single JavaScript object and add this object to the scratchpad instead of storing each value individually.

```javascript
// complex JavaScript object
var joe = {
    name: 'Joe Watkins',
    age: 45,
    job: 'Electrical Engineer',
    contact: ['760-555-1212', 'joew@fastereng.net'],
    vaction: { start: new Date("October 12, 2013"), days: 5 }
}

passUserToAnotherActivityViaScratchpad( joe );

function passUserToAnotherActivityViaScratchpad(usr) {
    workflow.scratchpad.user = usr;
}
```

Format JavaScript objects

When writing JavaScript objects to the scratchpad, maintain the standard JSON format and do not attempt to express the objects as strings.

Expressing a JavaScript object as a string is unnecessary and requires additional parsing. The workflow scratchpad can serialize and deserialize data to and from any JavaScript primitive, array, or object. This example shows objects stored incorrectly as strings.

```javascript
var user1 = "{ 'name':'Joe', age:12 }";
var user2 = "{ 'name':'Mary', age:42 }";
workflow.scratchpad.users = [user1, user2];
```
This second example shows the recommended way to store JavaScript objects.

```javascript
var user1 = { 'name':'Joe', age:12 }; 
var user2 = { 'name':'Mary', age:42 }; 
workflow.scratchpad.users = [user1, user2];
```

**Scriptable Java objects**

A scriptable Java object is any Java object that is accessible through JavaScript, such as GlideRecord or GlideDateTime.

Any workflow activity can access scriptable Java objects, but any scriptable Java object declared within an activity is available only to that activity. Do not store scriptable Java objects as scratchpad variables. Instead, store primitive data from the Java object that a follow-up activity can use to reconstruct the Java object, such as the object **Name** or unique **ID**.

**Forwarding a single-row GlideRecord object**

When forwarding a single row of data from a GlideRecord object, store the sys_id of that row on the scratchpad.

In this example, the sending activity instantiates a GlideRecord object from a passed Incident **Number**. The activity then updates the **Short description** of that incident and adds the sys_id of the incident to the scratchpad.

```javascript
// assume that 'incidentNumber' comes from a workflow input variable, activity variable, or other source.
// read the row into a glide record.
var grIncident = new GlideRecord('incident');
grIncident.get('number', incidentNumber);
grIncident.short_description = 'IMPORTANT: needs attention');
grIncident.update();

// write this row to the scratchpad for the next activity:
workflow.scratchpad.incidentId = gr.getUniqueValue();
```

The follow-up activity then uses the sys_id value passed in the `workflow.scratchpad.incidentId` scratchpad variable to instantiate a new GlideRecord object for that row.

```javascript
// re-read the incident in the follow-up activity: 
var grIncident = new GlideRecord('incident');
grIncident.get( workflow.scratchpad.incidentId );
```

**Forwarding a multi-row GlideRecord object**

When passing multiple rows from a GlideRecord object, pass the query used to generate the GlideRecord.

In this example, the sending activity writes enough information about a GlideRecord object to the scratchpad for the follow-up activity to recreate the GlideRecord object.

```javascript
function passJoesIncidentsToProcessingActivity() { 
  // first find Joe User 
  var joeUser = new GlideRecord('sys_user'); 
  if (!joeUser.get('name', 'Joe User')) { 
    workflow.debug('User: Joe User not found'); 
    return; 
  }
  
  // read the row into a glide record. 
  var grIncident = new GlideRecord('incident'); 
  grIncident.addQuery('caller', joeUser.getUniqueValue() );
  grIncident.query();
}```
sendQueryGR('joesIncidents', grIncident);
}

// send any filtered GR to another activity via scratchapd
function sendQueryGr(variableName, gr) {
    workflow.scratchpad[variableName] = grDescriptor(gr);
}

function grDescriptor(gr) {
    return {
        sys_id: gr.getUniqueValue(), // current row, if applicable
        table: gr.getTableName(),
        query: gr.getEncodedQuery()
    }
}

The follow-up activity then duplicates the original GlideRecord query using the scratchpad values.

// Re-query in the follow-up activity:
var gr = reloadQuery('joesIncidents');

function reloadQuery(variableName) {
    var grDescriptor = workflow.scratchpad[variableName];
    var gr = new GlideRecord(grDescriptor.table);
    gr.addEncodedQuery(grDescriptor.query);
    gr.query();

    // [Optional] seek to the same row
    while (gr.next()) {
        if (gr.getUniqueValue() == grDescriptor.sys_id)
            return;
    }
    workflow.debug(grDescriptor.table + ':' + grDescriptor.sys_id + ' - record not found');
}

Current variables

Current is the database GlideRecord that kicked off the workflow, either by association to the table in the Workflow properties table or by being associated with a catalog item.

Declaration

Variables in current are the columns that are defined in the dictionary.xml file that support the database record. There is no way at runtime to add variables to the current record.

Display

Variables of a current record are displayed on the Glide Forms and Lists throughout the applications that use them.
Access and Assignment

To access the variables or assign values to the variables within the workflow activity, get the value from the current record by referencing the Glide Element.

```
var myVarValue = current.getElement(nameOfTheField);
```

or

```
var myVarValue = current.fieldName;
```

or

```
current.variableName.setValue( "A Variable Value"");
```

Workflow events

The system employs two types of events: registered platform events and workflow events.

Registered platform events

Registered events are created in business rules and are used for such tasks as sending email notifications when records are inserted into the database. Workflow events are registered within workflows only and are not used anywhere else in the platform. Registered platform events can be triggered by a workflow for external use, but cannot be used within a workflow.

Workflow events

Workflow events follow different rules than platform events that are registered using the event registry. Platform events are entered into the Event Registration [sysevent_register] table and are available for platform processes to use. Workflow events are triggered exclusively for the workflow engine and are used only to direct the work of executing workflow contexts. When an event is registered in a workflow, it is attached to a currently executing activity in the registered_events column of the Workflow Executing Activity [wf_executing] table.

Workflow events also can be broadcast to a workflow from any scripting source that has access to the workflow context, such as a script include or a Run Script activity. In this case, the event, such as cancel, is passed to all records in the Workflow Executing [wf_executing] table for a specific context.

Whether by registry or by broadcast, an event is handled by the activity definition associated with the currently executing activity. Each activity definition comes with a set of handlers. For example, most activities come with onExecute, onCancel, and onUpdate event handlers. As an example of a more specific event, the Approval - User activity also comes with onDetermineApprovalState, which is specific to the work of the approval activity.

Multiple parallel events

A single workflow can have multiple event threads running concurrently, such as when a workflow has timers that overlap on separate workflow branches.

If any additional thread completes before the first thread, the system stores event information from the additional thread on the Workflow Queued Commands [wf_command] table. After the first thread completes, the system retrieves the information stored by the additional thread and proceeds through the workflow with the event information from each thread.

Workflow events in the base system

Several workflow events are available in the base system.
<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
<th>Purpose</th>
<th>To use</th>
<th>Source</th>
<th>Thread</th>
<th>Listeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>activityComplete</td>
<td>String value used by activity definitions to respond to the onActivityComplete event handler.</td>
<td>Informs records in the Workflow Executing Activity [wf_executing] table about the completion of other activities in the same workflow context.</td>
<td>If the activity is allowed to set the boolean value for wf_executing.notify_termination, then set the value to true (activity.notify_termination = true) during the onExecute event.</td>
<td>Workflow Engine, Process Terminations</td>
<td>Current thread, current mutex</td>
<td>Join activity</td>
</tr>
<tr>
<td>otherEvent</td>
<td>String value used by the Join activity to respond to an otherEvent.</td>
<td>Informs records in the Workflow Executing Activity [wf_executing] table about an otherEvent that has completed.</td>
<td>The Join activity transitions from n number of preceding activities. These preceding activities all create a wf_executing record, which causes a check to see if the record already exists. If the Join already exists, then the Join created by the executing transition sets the wf_executing record for deletion.</td>
<td>Join activity</td>
<td>Current thread, current mutex</td>
<td>Join activity, onOtherEvent event handler</td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Purpose</td>
<td>To use</td>
<td>Source</td>
<td>Thread</td>
<td>Listeners</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>timer</td>
<td>String value used by workflow activities to respond to a Timer activity that has expired.</td>
<td>Allows wf_executing records to be informed about a timer activity that has completed and has fired the timer event.</td>
<td>The <strong>Timer</strong> activity schedules a job that calls a script. The script calls fireEvent (wf_executing, timer).</td>
<td><strong>Timer</strong> activity via a scheduled job</td>
<td>Worker thread, private mutex</td>
<td><strong>Timer activity, onTimer event handler</strong></td>
</tr>
<tr>
<td>execute</td>
<td>String value used by workflow activities to respond to a Timer activity that has expired.</td>
<td>Informs a record in the wf_executing table with the initial state of <strong>Executing</strong> to proceed with its primary work.</td>
<td>The workflow engine, for each transition executed, creates an executing record with a state of <strong>Executing</strong>. Once created, the executing record is put in a queue for processing. For each item in the queue, the Rhino globals are established, the activity definition that drives the executing record is instantiated, and the <code>run()</code> function is called. When the state of a record is <strong>Executing</strong>, this function always calls <code>onExecute</code>.</td>
<td><strong>Workflow engine, via the WFAActivity Handler</strong></td>
<td>Current thread, current mutex</td>
<td>All activities, <strong>onExecute event handler</strong></td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Purpose</td>
<td>To use</td>
<td>Source</td>
<td>Thread</td>
<td>Listeners</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>execute (specific to Lock)</td>
<td>String value used by the Lock activity to respond to a waiting lock that is ready to make another attempt to obtain a specific lock. This execute is different than the previous execute because it is called on a separate thread, at specified intervals, and is treated as an outside event.</td>
<td>Informs a Lock activity schedules a job with a script that uses the workflow script include's fireEvent(wf_executing, 'execute') method.</td>
<td>Lock activity via a scheduled job</td>
<td>Worker thread, private mutex</td>
<td>Lock activity, onExecute event handler</td>
<td></td>
</tr>
<tr>
<td>determineApprovalState</td>
<td>String value used by approval activities to respond to a change in the overall approval status of the current record.</td>
<td>Informs Approval Coordinator both registers for the event and triggers the event. The child approvals have listeners that determine their approval state.</td>
<td>Approval Coordinator triggers the event during its onExecute event</td>
<td>Current thread, current mutex</td>
<td>Approval Coordinator, Approval - User, and Approval - Group all have onDetermineApprovalState event handlers. If the state is anything but Requested, the activity is considered finished, and the approval state (Approved, Rejected, Cancelled) is set to the wf_executing.result column</td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Purpose</td>
<td>To use</td>
<td>Source</td>
<td>Thread</td>
<td>Listeners</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>cancel (from within activity definitions)</td>
<td>String value used by workflow activities to respond to a request for cancellation.</td>
<td>Informs all wf_executing records in a context that the workflow is being canceled.</td>
<td>The <strong>End</strong> activity uses the global workflow.broadcastEvent('cancel') to interrupt the currently running wf_executing records. This changes the state of those records to <strong>Cancelled</strong>.</td>
<td>End activity</td>
<td>Current thread, current mutex</td>
<td>All activities, <strong>onCancel</strong> event handler</td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Purpose</td>
<td>To use</td>
<td>Source</td>
<td>Thread</td>
<td>Listeners</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>cancel (outside current context)</td>
<td>String value used by workflow activities to respond to a request for cancellation. This event is the same as the cancel event above and handled the same. However, the management of this event is subtly different. This event informs all workflow records in a context that the workflow is being canceled. The event is managed via the <code>onCancel</code> event handler of each executing activity definition, but the event is called differently. In particular, the call to cancel from outside an activity definition is blocked by the current mutex. This is a significant difference in that the event does not interrupt a currently executing activity that is still operating within the parameters of the.</td>
<td>Any script can call cancel on a known executing context via the workflow script include. For example, var w = new Workflow(); w.cancel(context); //where context is a GlideRecord of the context to be canceled.</td>
<td>Any script include, scheduled job, UI action, or other source</td>
<td>Current thread, private mutex</td>
<td>All activities, <code>onCancel</code> event handler</td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Purpose</td>
<td>To use</td>
<td>Source</td>
<td>Thread</td>
<td>Listeners</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>stop (see End activity)</td>
<td>The End activity checks for this event.</td>
<td>If the stop event is the current event, then the cancel operation of the End activity is bypassed.</td>
<td>Only in the End activity.</td>
<td>Any script can trigger or broadcast the stop event via a script include or workflow Run Script activity</td>
<td>Current thread, current mutex</td>
<td>This event is used by the End activity to exclude the Cancel activity and allow a workflow to end, even if canceled.</td>
</tr>
<tr>
<td>listener</td>
<td>String value that the workflow (subflow) activity triggers as an event.</td>
<td>When a main workflow calls a subflow, the workflow keeps the ID of the subflow's context in the scratchpad. When the subflow is complete, it triggers the listener event via a business rule.</td>
<td>The listener event is passed to the parent context on completion of a subflow and is managed by the onListener action of the workflow activity.</td>
<td>Business rule that is triggered by the update of a workflow that has a parent</td>
<td>Current thread, current mutex</td>
<td>This event is used by a subflow to inform its parent flow that it is complete. The parent workflow will react to this event and continue.</td>
</tr>
<tr>
<td>probe_complete</td>
<td>String value triggered in the workflow by an Orchestration activity indicating that the MID Server has completed a task.</td>
<td>The probe_complete event is triggered from Orchestration sensor processors via the workflow helper function handleEventById.</td>
<td>Event used to restate a workflow that is waiting for the MID Server to process a task or activity</td>
<td>Orchestration activities</td>
<td>Worker thread, private mutex</td>
<td>Orchestration activities</td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Purpose</td>
<td>To use</td>
<td>Source</td>
<td>Thread</td>
<td>Listeners</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>pause</td>
<td>String value sent to a workflow from an SLA to pause the <strong>Timer</strong> activity.</td>
<td>When an SLA is paused, the SLA workflows must be paused if there is a timer running.</td>
<td>Use is exclusive to the SLA timer</td>
<td>SLA</td>
<td>Business rule thread, private mutex</td>
<td><strong>Timer</strong> activity</td>
</tr>
<tr>
<td>resume</td>
<td>String value used by the <strong>Timer</strong> activity to resume a paused timer (see pause).</td>
<td>When an SLA is resumed, the SLA workflows must be resumed as well.</td>
<td>Use is exclusive to the SLA timer.</td>
<td>SLA</td>
<td>Business rule thread, private mutex</td>
<td><strong>Timer</strong> activity</td>
</tr>
</tbody>
</table>

*Glide events relative to workflows*

Workflow uses several Glide events.

**Table 1235: Workflow Glide events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
<th>Purpose</th>
<th>To Use</th>
<th>Source</th>
<th>Thread</th>
<th>Listeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>Global event set upon the insert of a GlideRecord that causes the script engine, and through that the workflow engine, to wake up.</td>
<td>Starts workflows that are associated with the current GlideRecord either by reference, as in request items and SLA timers, or by conditions associated with the GlideRecord's table.</td>
<td>There is no explicit customer-facing use for this in a workflow. It is part of the Glide engine, and with this event, the only thing workflows can do is start. Workflows can also be started manually using a script.</td>
<td>Workflow Engine, RunEngine</td>
<td>Current thread, current mutex</td>
<td>User action of insert</td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Purpose</td>
<td>To Use</td>
<td>Source</td>
<td>Thread</td>
<td>Listeners</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>Update</td>
<td>Global event set upon the update of a GlideRecord that causes the script engine, and through that the workflow engine, to wake up.</td>
<td>Looks to the Workflow Context [wf_context] table to find running workflows that are associated with the current GlideRecord by document ID.</td>
<td>There is no explicit customer-facing use for this in a workflow. It is part of the Glide engine, and with this event, the only thing workflows can do is advance through the next set of transitions.</td>
<td>Workflow Engine, RunEngine</td>
<td>Current thread, current mutex</td>
<td>User action of update of a GlideRecord</td>
</tr>
<tr>
<td>Delete</td>
<td>Global event set upon the delete of a GlideRecord that causes the script engine, and through that the workflow engine, to wake up.</td>
<td>Looks to the Workflow Context [wf_context] table to find running workflows that are associated with the current GlideRecord by document ID.</td>
<td>There is no explicit customer-facing use for this in a workflow. It is part of the Glide engine, and with this event, the only thing workflows can do is advance through the next set of transitions.</td>
<td>Workflow Engine, RunEngine</td>
<td>Current thread, current mutex</td>
<td>User action of delete of a GlideRecord</td>
</tr>
</tbody>
</table>
### Workflow event-specific functions

There are several functions that relate specifically to workflow events.

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
<th>Purpose</th>
<th>To Use</th>
<th>Source</th>
<th>Thread</th>
<th>Listeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Global event set upon the query of the Glide database that causes the script engine, and through that the workflow engine, to wake up.</td>
<td>Looks to the Workflow Context [wf_context] table to find running workflows that are associated with the current GlideRecord by document ID.</td>
<td>There is no explicit customer-facing use for this in a workflow. It is part of the Glide engine, and with this event, the only thing workflows can do is advance through the next set of transitions.</td>
<td>Workflow Engine, RunEngine</td>
<td>Current thread, current mutex</td>
<td>User action of query of a GlideRecord</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Purpose</td>
<td>Use</td>
<td>Thread</td>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>registerForEvent</td>
<td>Function in the workflow environment that writes events represented as</td>
<td>The workflow events are just strings. When an activity that has</td>
<td>The global variable workflow that is available to all Workflow Activity</td>
<td>Current thread, current mutex</td>
<td>Global variable workflow</td>
<td></td>
</tr>
<tr>
<td>(eventName)</td>
<td>strings to the wf_executing.registered_events field.</td>
<td>has registered for an event executes, a comma delimited set of events</td>
<td>records is the source of the call. For example, from inside a Run</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>is stored with the Workflow Executing Activity [wf_executing] record.</td>
<td>Script activity, a designer can write:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>When the event is triggered in the workflow context, the wf_executing</td>
<td>workflow.registerForEvent('myEventName');</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>table looks for all executing records that contain the string that</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>represents the event in the wf_executing.registered_events field</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Purpose</th>
<th>Use</th>
<th>Thread</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>unRegisterForEvent</td>
<td>Function in the workflow environment that removes a string value representing an event that has been written to the wf_executing.registered_events field.</td>
<td>The workflow events are just strings that are written to the wf_executing.registered_events field. When an activity unRegisters for an event, the comma-delimited set of events stored with the Workflow Executing Activity [wf_executing] record is searched, and if that string is found, it is removed.</td>
<td>The global variable workflow that is available to all Workflow Activity [wf_activity] records is the source of the call. For example, from inside a <strong>Run Script</strong> activity, a designer can write: <code>workflow.unRegisterForEvent('myEventName');</code></td>
<td>Current thread, current mutex</td>
<td>Global variable workflow</td>
</tr>
<tr>
<td>fireEvent</td>
<td>Function in the workflow environment that examines the contents of the wf_executing.registered_events field, comparing its contents to the eventName passed in.</td>
<td>The workflow events are just strings that are written to the wf_executing.registered_events field. When fireEvent(eventName) is called by a workflow activity, the workflow engine queues up any executing records that contain the string in the registered field.</td>
<td>The global variable workflow that is available to all Workflow Activity [wf_activity] records is the source of the call. For example, from inside a <strong>Run Script</strong> activity, a designer can write: <code>workflow.fireEvent('myEventName');</code></td>
<td>Current thread, current mutex</td>
<td>Global variable workflow</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Purpose</td>
<td>Use</td>
<td>Thread</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
<td>-----</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>fireEvent (eventRecord, eventName)</td>
<td>Function in the workflow environment that sends an event to a specific Workflow Executing Activity [wf_executing] record. The eventRecord is a GlideRecord of the type wf_executing.</td>
<td>This event call expects an onMyEvent event handler in the activity represented in the event record (Workflow Executing Activity [wf_executing] table). When fireEvent (eventRecord, eventName) is called by a workflow activity, the workflow engine queues up the specific executing record with that event and passes the event into the activity definition for the on&lt;eventName&gt; handler to manage. This event is queued up in its own mutex, so the current queue completes before this event is processed.</td>
<td>The workflow script include contains the call for this. For example, from inside a Run Script activity, a designer can write: var w = new Workflow(); w.fireEvent (executing, eventName);</td>
<td>Current thread, current mutex</td>
<td>Workflow script include</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Purpose</td>
<td>Use</td>
<td>Thread</td>
<td>Source</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>fireEvent (eventRecordSysId, eventName)</td>
<td>Function in the workflow environment that sends an event to a specific Workflow Executing Activity [wf_executing] record. The eventRecordSysId is the sys_id of a GlideRecord of the type wf_executing.</td>
<td>This is the same as the fireEvent above, except that it accepts an ID and returns the Workflow Executing Activity [wf_executing] record.</td>
<td>The Workflow script include contains the call for this. For example, from inside a Run Script activity, a designer can write: <code>javascript var w = new Workflow(); w.fireEvent(executing, eventName); </code></td>
<td>Current thread, current mutex, current mutex</td>
<td>Workflow script include</td>
</tr>
<tr>
<td>fireEvent (eventRecordSysId, eventName, optionalJSONObject)</td>
<td>Function in the workflow environment that sends an event to a specific Workflow Executing Activity [wf_executing] record. The eventRecordSysId is the sys_id of a GlideRecord of the type wf_executing.</td>
<td>This is the same as the fireEvent above, except that it accepts a JSON object as a third parameter. This object can specify any data expressible as JSON. You can also specify additional functionality when creating a workflow activity.</td>
<td>The Workflow script include contains the call for this. For example, from inside a Run Script activity, a designer can write: <code>javascript var w = new Workflow(); w.fireEvent(executing, eventName, JSONObject); </code></td>
<td>Current thread, current mutex</td>
<td>Workflow script include</td>
</tr>
<tr>
<td>broadcastEvent (contextId, eventName)</td>
<td>Function in the workflow environment that sends an event to all currently running Workflow Executing Activity [wf_executing] records in a specified context, regardless of their state.</td>
<td>This is the same as the fireEvent above, except that it accepts an ID and returns the Workflow Executing Activity [wf_executing] record.</td>
<td>The Workflow script include contains the call for this. For example, from inside a Run Script activity, a designer can write: <code>javascript var w = new Workflow(); w.broadcastEvent(contextId, eventName); </code></td>
<td>Current thread, current mutex</td>
<td>Workflow script include</td>
</tr>
</tbody>
</table>
### Event-specific workflow activities

The following workflow activities trigger events.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Purpose</th>
<th>Use</th>
<th>Thread</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>broadcastEvent ((eventName))</td>
<td>Function in the workflow environment that sends an event to all currently running Workflow Executing Activity ([wf_executing]) records in the current context, regardless of their state.</td>
<td>This should not be confused with broadcastEvent above. This event is only available to current Workflow Executing Activity ([wf_executing]) records.</td>
<td>This is available only through the global workflow variable of the current context. The following is an example of its use from within an activity definition's script: <code>workflow.broadcastEvent(eventName)</code></td>
<td>Current thread, current mutex</td>
<td>Global variable workflow</td>
</tr>
</tbody>
</table>
Table 1237: Event-specific workflow activities

<table>
<thead>
<tr>
<th>Activity Name</th>
<th>Description</th>
<th>Purpose</th>
<th>To Use</th>
<th>Source</th>
<th>Thread</th>
<th>Listeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Event</td>
<td>Requires an event from the event registry rather than a workflow event. This activity is located in the Notification category of the workflow tree.</td>
<td>Fires the notification event specified in the Workflow Activity table.</td>
<td>1. Navigate to System Policy &gt; Events &gt; Registry 2. Create an event. 3. Navigate to System Policy &gt; Templates and create an email template. 4. Navigate to System Policy &gt; Email &gt; Notifications. 5. Create a new notification that is triggered by the event you created and sends the template you created. 6. On the workflow canvas, drag the Create Event activity onto the canvas and associate</td>
<td>Event Registry</td>
<td>Event thread</td>
<td>On the notification thread, outside of workflow</td>
</tr>
</tbody>
</table>

© 2017 ServiceNow. All rights reserved.
<table>
<thead>
<tr>
<th>Activity Name</th>
<th>Description</th>
<th>Purpose</th>
<th>To Use</th>
<th>Source</th>
<th>Thread</th>
<th>Listeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait for WF Event</td>
<td>Listens for workflow events, as described in the Workflow Events in the Base System table, and only within the current context. This activity is located in the Conditions category of the workflow tree.</td>
<td>Waits for another transition branch of the current context to trigger an event.</td>
<td>Takes an event name as input. When the activity is executed, the specified event name is added to a string array stored in the registered_events column. The values in this column are in a list of all events the activity waits for when it is executing.</td>
<td>The Wait for WF Event activity has a generic onUnhandledEvent that tests the current event against the value that was passed to the variable. If they match, the Wait for WF Event moves the workflow forward.</td>
<td>Triggered in the current thread or from a script include Event</td>
<td>The onUnhandledEvent of the Wait for WF Event activity</td>
</tr>
</tbody>
</table>

Workflow stages

Workflows can indicate workflow progress by updating any field designated as a stage field.

For example, the Incident [incident] table has an Incident state field that indicates progress, but the service catalog uses the Stage field.

To indicate the workflow's progress through the possible stage values, the interface updates the Stage field selected in the workflow properties. Available fields depend on the table used by the workflow. If the field provides a choice list, then the choices are available as stage values for the workflow. If the field is a workflow field, it displays an icon to indicate the workflow's progress, as with the Service Catalog's Stage field.
After stages are added to the workflow, they can be assigned to each workflow activity. If an activity with an assigned stage is encountered when the workflow runs, the workflow engine assigns the stage to the record associated with the workflow context.

For workflows that use the Requested Item [sc_req_item] table, the stage field is automatically set to the Stage field of the table and cannot be changed. The stage state displayed for a workflow running on the Requested Item table is based on the state of the workflow activities. If an activity is active, then the stage is shown with the state of In progress. If an activity is in the Pending or Completed state, the stage reflects this state.

Workflow stage sets

You can create stage sets that are named groups of stages commonly used together. These stage sets can be used in any number of workflows on different tables.

A single stage set usually represents a process, such as the stages required to display the progress of a service catalog request. You can also assign a stage set as the default set for workflows on specific tables.

In an active context, workflow stages provide summary-level feedback about the progress of a workflow. Stage icons display the status of each activity as it is being driven by a workflow. The system updates a Stage field, defined in the workflow properties, to indicate the workflow's progress through the possible stage values.

Updates to workflow stage values

During an upgrade, the system makes changes to the Value field of records in the Workflow Stage [wf_stage] table.

The following changes are made:

- All entries are made lowercase.
- All spaces and special characters are replaced with underscores.

These changes support internationalization of choice lists and enable workflows to display translated text in the Stage field.
Workflow validation

Workflow validation examines different characteristics of a workflow to locate issues that might prevent the workflow from being published or cause it to fail.

Validation prevents workflows with critical flaws from executing and resulting in an unstable or incomplete state. There are a number of validators in the base system that notify workflow designers of potential problems. For example, multiple End activities, disconnected transitions, incorrect table references, missing subflows, and dependencies affected by update sets. A workflow validation report displays the results from each validator, including a message explaining what was found. The system automatically validates a workflow when you publish it. You can also run validation on a workflow directly from controls in the Workflow Editor.

Validate a workflow from the workflow editor

You can manually validate a workflow from the workflow editor.

Role required: workflow_admin

Running a workflow on a new node automatically attempts to validate the workflow. If validation is successful, the system updates the workflow version record to indicate the workflow has been validated and marks the record as updated by the user who ran the workflow.

1. Open the workflow to validate in the workflow editor.

   When the workflow is loaded, the workflow validator icon appears in the toolbar.

2. Click the validator icon to run a series of validation tests on the current workflow version and generate a report.
Validate a workflow from the Workflow Version form

You can generate a workflow validation report from the Workflow Version form.

Role required: workflow_admin

1. Navigate to Workflow > Administration > Workflow Versions.
2. Select a workflow to validate.

Highlighting critical errors

The graphical workflow editor highlights critical errors when a workflow is loaded.
In this example, a subflow is missing and is not available to the parent workflow for the current user. The graphical workflow editor indicates the error when the parent workflow loads by highlighting (in red) the activity that calls the subflow. To correct the error in the parent workflow, click the validate icon in the header bar and inspect the error description in the validation report.

![Diagram of workflow with missing subflow]

**Figure 863: Validate missing subflow**

**Validations at publishing**

If you attempt to publish an invalid workflow or a workflow with potential problems, the system displays an error message and blocks the operation, if necessary.

When validation error messages appear, click the validate icon in the graphical workflow editor to see the error report.

**Validation warning**

A validation warning notifies you that a potential problem exists in a workflow but permits you to publish the workflow. Validation warnings appear when:

- You edit and then attempt to publish a workflow that is included as a subflow in another workflow. The system cannot determine how your changes will affect the parent workflow and alerts you of the relationship.
- A workflow activity uses a different table than the table assigned to the workflow. The system alerts you of the potential conflict.
Publish Confirmation

Validation warning. Click the validation button in the toolbar to get details:

Validate Summary - Workflow version contains Warnings - Total checks performed:15 (Info:13, Warn:2, Critical:0)

Publish this workflow with warnings?

Cancel  OK

Figure 864: Validation warning

Validation failure

A validation failure notifies you that a critical error has occurred in the workflow that prevents you from publishing the workflow. An example of a critical error is a missing subflow.

Figure 865: Validation failure

Workflow validation report

The validation report summarizes the results of each separate workflow validation.

Validators display three notification levels: CRITICAL, WARN, and INFO. The designer can publish a workflow that returns WARN or INFO level validation, but not a workflow that returns an overall validation level of CRITICAL.
Figure 866: Workflow validation report

Header summary

The header of the validation report summarizes the entire validation run against the specified workflow.

- **Validate Summary**: The overall score reflects the most severe notification level encountered during the validation.
- **Total checks performed**: The total number of validations run is also broken down to show the number at each notification level.
Report columns

The body of the report displays the results of each individual validation check that was performed. The columns are **Type**, **Level**, and **Message**. You can sort and filter these columns as you would any list.

Levels

<table>
<thead>
<tr>
<th>Name</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Provides information about the current workflow version. An example of an information level message is one that names the lowest common table in the workflow. Workflows at this validation level are considered valid and publishable.</td>
</tr>
<tr>
<td>Warning</td>
<td>Alerts the user that the validator detected anomalies in the workflow that might compromise its executability. An example of a warning level message is one that alerts you to a missing activity input transition. Workflows at this validation level are considered valid and publishable.</td>
</tr>
<tr>
<td>Critical</td>
<td>Names a workflow element containing a critical error that prevents the workflow from executing successfully. Examples of this are missing or invalid subflows and missing transitions. Workflows at this validation level cannot be published or run in production.</td>
</tr>
</tbody>
</table>

Message

The validation message provides a detailed description of the results, including table names, update sets, and other specifics.

Workflow validator

ServiceNow offers several workflow validators for workflow designers to test their workflows.

This page lists all available workflow validators. See Workflow validation on page 3589 for information on using workflow validators and Workflow validation report on page 3592 to see the type of information that is returned.

<table>
<thead>
<tr>
<th>Hanging Workflows</th>
<th>Update Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify workflow design decisions that can result in a hanging workflow</td>
<td>Identify related artifacts being moved in different update sets</td>
</tr>
<tr>
<td>ValidateTransitionOut on page 3595</td>
<td>ValidateUpdateSetDependencies on page 3606</td>
</tr>
<tr>
<td>ValidateTransitionIn on page 3596</td>
<td>ValidateUpdateSetParentDependencies on page 3608</td>
</tr>
</tbody>
</table>
Table 1240: Workflow termination and external dependencies

<table>
<thead>
<tr>
<th>Unexpected Workflow Termination</th>
<th>External Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify workflows that can end unexpectedly</td>
<td>Identify external artifacts that have potential workflow dependencies</td>
</tr>
<tr>
<td>ValidateSingleEnd on page 3603</td>
<td>ValidateParentFlow on page 3603</td>
</tr>
</tbody>
</table>

Table 1241: Workflow conflicts

<table>
<thead>
<tr>
<th>Workflow Properties Conflicts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify workflow design decisions that conflict with workflow properties</td>
</tr>
<tr>
<td>ValidateLowestCommonTable on page 3601</td>
</tr>
<tr>
<td>ValidateTableChange on page 3602</td>
</tr>
</tbody>
</table>

**ValidateTransitionOut**

The **ValidateTransitionOut** validator finds activity conditions with no exit transitions.

**Validation summary**

- **Risk:** Activity conditions might not transition to the next activity, which could cause the workflow to hang.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** All conditions have transitions.
- **Invalid Result:** Invalid
- **Invalid Message:** This workflow contains <condition count> activity conditions without an output transition.
- **Suggested Action:** If this is a conscious design decision, there is no corrective action. Otherwise, find the condition cited in the validator and add an appropriate transition to the next activity.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** None

**Troubleshooting**

Design choices made when creating a workflow on the canvas might legitimately use an activity without an exit condition. In the first example, the **Notification** and **Timer** activities both execute at the start of...
the workflow. The Timer is the entity that decides when the workflow ends. In this situation, executing the Notification, but not transitioning away, keeps the design simple and adds no risk. The validator finds and reports the missing transition from the Notification activity as a Warning that the designer can ignore.

Figure 867: Condition with no valid transition

In the second example, the Notification activity has no exit transition. The designer missed this because of the layout. The transition from the Timer activity passes behind the Notification activity and appears to connect the exit from the Notification activity to the End. In workflows with more than 10 or 15 activities, it might be difficult to see all the transitions clearly. This workflow’s designer intended for the Notification activity to transition to the End.

Figure 868: No condition out

This validator directs the designer to the specific activity and condition that does not have an exit transition. The designer then makes the decision whether or not to respond to the warning.

ValidateTransitionIn

The ValidateTransitionIn validator finds activities that do not have inbound transitions and cannot execute in the workflow.
Validation summary

- **Risk:** Activities that do not have inbound transitions have no means of being executed in the workflow. If other logic builds from these activities, the workflow could hang, with no means of moving forward.
- **Purpose:** Find activities that do not have inbound transitions.
- **Severity Level:** Warning.
- **Valid Result:** Valid.
- **Valid Message:** All activities in this workflow have at least one inbound transition.
- **Invalid Result:** Invalid.
- **Invalid Message:** This workflow contains <activity count> activity conditions without an input transition.
- **Suggested Action:** Either remove the activities to reduce confusion and make the workflow easier to understand or provide the appropriate inbound transition.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** None

Troubleshooting

Although the activities detected in this validator do no harm to the execution of the workflow, the designer needs to know that there are activities on the canvas that cannot execute and serve no purpose, particularly in a production system. This error is typically caused by a visual misinterpretation of the drawing.

This example workflow appears to be perfectly valid. Under closer inspection, however, we see that the transition from **Set Values** does not stop at **Approval - User** (there is no arrow end). Instead, that transition goes directly to **End**. As a result, **Approval - User** cannot execute. Because there is an activity in the workflow based on the approval's condition routing, it appears that the designer intended for the approval to execute and that a correction is needed.

![Diagram of workflow with transition issue](image)

**Figure 869: TransitionIn invalid**

© 2017 ServiceNow. All rights reserved. 3597
This validator directs the designer to the specific activity and condition that has no inbound transition. The designer should establish a transition or remove the activity.

**ValidateDanglingTransition**

The **ValidateDanglingTransition** validator finds and reports any transitions that do not terminate on an activity.

---

**Note:** These transitions are not drawn on the workflow canvas, but are still present in the database.

---

**Warning:** This is a critical error that prevents a workflow from running.

---

**Validation summary**

- **Risk:** A workflow with dangling transitions will silently hang a workflow with no recovery options.
- **Severity Level:** Critical
- **Valid Result:** Valid
- **Valid Message:** Valid
- **Invalid Result:** Invalid
- **Invalid Message:** Invalid
- **Suggested Action:** Remove or connect the offending transition. Get the source activity name from the validation report details and resolve the issue. Then, run the validation again to test your changes.
- **Publishable:** No
- **Runnable:** No
- **Related Information:** None

---

**Troubleshooting**

On rare occasions, the destination of a workflow transition becomes null. The workflow canvas shows no evidence of the transition, but at run time, the workflow hangs when it encounters one of these dangling transitions. If the **ValidateDanglingTransition** validator reports this condition at publishing time, it blocks the publication action until the issue is resolved. If this condition is detected on a runtime check, the workflow is not allowed to execute against a current record's transaction. Instead, the system adds a critical log entry detailing the activity with the faulted transition to the current record's workflow context. To enable the workflow to execute on the next appropriate transaction, remove the faulted transition from the workflow model.

To find and remove the faulted transition:

1. Make note of the workflow version and activity that contains the faulted transition as indicated in the validator details.
2. Navigate to **Workflow > Administration > Workflow Version**.
3. In the list of workflow versions, select the workflow that has the faulted transition.
4. On Workflow Version form, add the workflow activities related list. Click the menu icon, select **Configure > Related Lists**, move **Workflow Activity-->Workflow Version** from the **Available** list to the **Selected** list, and click **Save**.
5. In the **Workflow Activities** related list, select the activity cited in the validator.
6. In the Workflow Activity form, view the **Workflow Transitions** section or tab and identify the transition in that list that has no value or a null value in the **To** column.
7. Delete this transition.
8. Return to the workflow version and re-run the validation check.

The **Critical** warning should disappear. The workflow should execute as expected on the next appropriate transaction.

**ValidateSubflows**

The **ValidateSubflows** validator detects any workflows included as subflows that are either inactive, deleted, or not available as a published workflow for the current user.

Any of these conditions cause the workflow to hang when the workflow activity in the main flow is encountered.

![Warning](icon) **Warning:** This is a critical error that prevents a workflow from running.

**Validation summary**

- **Risk:** A parent workflow that transitions to a deleted subflow hangs indefinitely, with no recovery options.
- **Severity Level:** Critical
- **Valid Result:** Valid
- **Valid Message:** This workflow contains <count> valid subflows.
- **Invalid Result:** Invalid
- **Invalid Message:** This workflow contains <invalid count> invalid subflow(s) of <total subflow count> total subflows.
- **Suggested Action:** Remove the link in the parent workflow to the questionable subflow, examine the subflow to ensure that it is valid and published, or that it is checked out to the current user. After making the correction to the state of the subflow, run the validation again to test your changes.

- **Publishable:** No
- **Runnable:** No
- **Related Information:** *Workflows used as subflows* on page 3552

**Troubleshooting**

When a workflow runs, regardless of whether it is a subflow or a main flow, the script engine determines which version of a workflow should execute, given the current user and workflow conditions. When a workflow is checked out by the same user who is running the workflow, the checked out version is the version that executes. If the user is not the same person who has the workflow checked out, the published version of the workflow executes. If there is no published workflow, no workflow runs.

One scenario addressed by the **ValidateSubflows** validator is when a workflow:

- Is checked out to User A.
- Is a subflow in a parent workflow being run by User B.
- Has no published alternative to the subflow being run by User B.

When this occurs, the parent workflow runs to the execution of the unpublished subflow and then hangs at that activity, with no means to transition forward. Main flows that encounter this condition in a subflow are not permitted to execute against a current record's transaction. Instead, a critical log entry detailing the subflow's state is added to the current workflow's Workflow Context record. To correct the problem, remove the subflow from the main flow, or publish the subflow so it is available to User B. This allows the workflow to execute on the next appropriate transaction.
Another scenario addressed by the **ValidateSubflow** validator is when a workflow:

- Is a subflow in a parent workflow being run by any user.
- Has no published alternative to the subflow, because the workflow has been deleted or all versions of the workflow are unpublished or inactive.

**Note:** You cannot delete a from a list or form workflow that is a subflow. However, you can create one of these unstable conditions with advanced scripting, SQL options, or incomplete update sets that contain main flows, but not the referenced subflows. When troubleshooting a workflow that triggered this validator, consider the history of the subflow while assessing the error condition.

**Determine whether a workflow can run**

To determine if a workflow can run:

1. In the navigation filter, enter `wf_workflow.list`.
   A list of workflows appears.
2. Open one of the workflows from the list.
3. In the **Versions** related list, check for all of the following:
   a) There is only one workflow version in a state of Checked out and Checked out by.
   b) There is only one version and it is not checked out. This version must be both Active and Published. (You may need to personalize the list and add the **Active** column.)
   c) If there are multiple versions, only one is Published.

These checks determine the only two conditions under which a workflow can run:

- A checked out version of a workflow is available for the user who has it checked out.
- A valid, published version of a workflow is available for all users who have permission to run the workflow.

Main flows containing subflows that do not meet one of these two conditions are not permitted to execute against a current record's transaction. Instead, a critical log entry detailing the subflow's state is added to the current workflow's Workflow Context record. To enable the workflow to execute on the next appropriate transaction, remove the subflow from the main flow or modify the subflow's published and active states.

**ValidateScriptForCurrentDotUpdate**

The **ValidateScriptForCurrentDotUpdate** validator finds workflow activities with scripts that use the `current.update()` function.

Calling `current.update()` causes significant performance delays in transaction processing and might cause an instance to hang.

**Validation summary**

- **Risk:** At best, a workflow that uses `current.update()` in scripts experiences degraded performance. In the worst case, the workflow enters an infinite, recursive loop that crashes the server.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** The JavaScript in this workflow has no instances of 'current.update()'.
- **Invalid Result:** Invalid
- **Invalid Message:** This workflow uses 'current.update()' in <count of current.update references> JavaScript statements.
Problems with current.update() in workflow scripts

A workflow initiates execution in one of these ways:

- Script Engine: If a workflow is assigned to a specific table, and given a run condition, the workflow runs on INSERT.
- Script: Any business rule, script include, background script, or client script can initiate a workflow using the workflow script include and calling startFlow().

The workflow engine initiates a workflow based on the matched criteria of the current record being inserted. The transaction for current is managed by the script engine and not the workflow. Workflows that progress on the update() of the current record are not invoked via the workflow engine, but as a call from either a client script or business rule. In either case, the script engine is invoked, and the current record is put in memory. Edits and modifications to any current fields are made and are available to other activities and scripts that are executed in the same transaction.

When appropriate, other engines that run in sequence with the workflow engine, such as the business rule engine or field normalization, are invoked against the same current record transaction. Any changes made to current through these scripts and activities modify the record in memory. These changing values are available for reference in any other transactions called from activities and scripts in the same INSERT transaction. When all expected changes are executed, the current record is inserted.

When one of these scripts calls current.update() on a record that has yet to be inserted, the action forces an unnecessary and error-prone database transaction. If a record is not yet in the database, it cannot be updated. Business rules that trigger on update() on a record that is in the process of being inserted can cause a very unstable and potentially infinite looping condition.

Troubleshooting

This validator detects the use of current.update() in any of the editable script fields. Do not call current.update() from within a workflow script. In the event of an INSERT or UPDATE of current, the changes made to current are available to all scripts executing in the same transaction, and the script engine stores all changes in the database. Leave the update of current to the engine. Use the scripts only for setting and referencing the current field values.

ValidateLowestCommonTable

The ValidateLowestCommonTable validator reports the lowest table in the Glide hierarchy that the workflow uses.

For example, the Requested Item [sc_req_item] table is the lowest table in a workflow containing a Catalog Task activity. This information is significant to a designer who wants to change the table against which an existing workflow runs after adding activities to the canvas.

Note: This validator provides information only. It does not indicate an error or warning condition.
Validation summary

- **Risk:** This validator informs only and has no risk associated with it.
- **Severity Level:** Data/Information
- **Valid Result:** Valid
- **Valid Message:** The lowest common table in this workflow is `<table_name>`.
- **Invalid Result:** N/A (informational only)
- **Invalid Message:** N/A (informational only)
- **Suggested Action:** None
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** *Tables and classes* on page 1417, *Workflow activities* on page 3676

**ValidateTableChange**
The *ValidateTableChange* validator reports any activities in the workflow that are invalid given the table associated with the workflow version.

For example, a workflow version that is associated with the Change Request [change_request] table but has a *Catalog Request* activity on the canvas is invalid, since the activity is not compatible with the selected table.

Validation summary

- **Risk:** If the current record at runtime does not originate from the table specified by the lowest common table, the activities for the lowest common table cannot set specific values.
- **Severity Level:** Warning if the table that is associated with a workflow is higher in the table hierarchy than the lowest common table required for the workflow activities.
- **Valid Result:** Valid
- **Valid Message:** All activities are valid for the newly selected table
- **Invalid Result:** Invalid Activity
- **Invalid Message:** This workflow contains <count of invalid activities> invalid activities for the newly selected table.
- **Suggested Action:** Make one of these changes:
  - Change the workflow to not require the activities associated with the lowest table reported.
  - Modify the workflow to use a table that contains the lowest common table in its hierarchy.
  - Ensure that the current record meets the requirements of the at-risk activities.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** *Workflow activities* on page 3676

Troubleshooting

Navigate to *Workflow > Administration > Activity Definitions.* Note the *Table* column in the list. Each activity that is not global is associated with a table.

When you select a table in the Workflow Properties form, the activity menu presents only activities that are compatible with the selected table. Associating a table with a workflow activity enables the system to make certain decisions about the activity that affect value comparison, condition routing, or the setting and getting of field values in the current record.
When a workflow is set to the Global [global] table, it is possible that the process executing the workflow ensures that the current record meets the requirements of the activities identified by this validator. In that case, the activity still works as expected. If that assurance cannot be guaranteed by the user process, do not use the activity identified by the validator without assigning a table that meets the requirements of all the activities on the canvas.

This validation check ensures that the table specified by all the activities in a particular workflow is included in the hierarchy of the table selected for the workflow.

**ValidateParentFlow**
The **ValidateParentFlow** validator reports any workflows that use the workflow as a subflow.

**Validation summary**

- **Risk:** There is no risk in a workflow being a subflow. This is only a warning that other workflows are at risk from dramatic changes to a subflow.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** Currently `<workflow version name>` is not a subflow.
- **Invalid Result:** None
- **Warning Message:** This workflow version (`<workflow version name>`) is required as a subflow in `<workflow version count>` other workflows.
- **Suggested Action:** Exercise caution when modifying things like input parameters and return values to assure that parent workflows are not adversely affected.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** *Workflows used as subflows* on page 3552

**Troubleshooting**

The system warns the workflow designer at publishing time and during validation that a workflow is a subflow. This warning reminds the designer that changes to the current workflow have the potential to affect workflows cited in the validation report or other workflows already running in production. Regardless of how simple the change to a workflow that is a subflow, thoroughly test all parent flows cited in this validation report before publication.

When a workflow is a subflow, changes that can cause it to become invalid include:

- Changing the data types of *input variables*. Verify that all parent workflows cited in this validation report can pass the correct value type.
- Adding input variables. Verify that all parent workflows cited in this validation report are able to pass all variable values into the subflow.
- Changing or removing the return value of a workflow. Verify that any changes to the return value of a workflow are compatible with the requirements of all the parent flows.
- Changing the table on which the workflow runs. Verify that the table selection is compatible with all parent flows.

**Note:** To delete a workflow that is a subflow, first remove the dependency by removing the subflow from all parent flows cited in this validation report. After the dependencies are cleared, a user with the proper role can delete the subflow.

**ValidateSingleEnd**
The **ValidateSingleEnd** validator finds and identifies multiple *End* activities in a single workflow.
Multiple **End** activities in a workflow might be intentional and have no affect on the workflow, or might be a mistake that the designer needs to correct.

**Validation summary**

- **Risk:** If the execution paths to the *End* activities are not mutually exclusive, then the first *End* encountered completes the workflow and cancels all other executing activities.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** This workflow contains 1 End activity.
- **Invalid Result:** Invalid Activity
- **Invalid Message:** This workflow contains <count of ends> End activities.
- **Suggested Action:** Remove extraneous *End* activities that are not intended as part of the design.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** None

**Troubleshooting**

As soon as an **End** activity is encountered in the workflow, the workflow completes even if there are other viable execution paths leading to a second *End* activity that is still executing. Those executing activities are canceled as part of the *End* activity's clean up actions. Therefore, the results of designing workflows with multiple *Ends* must be carefully considered.

In the case of large workflows, it is often more intuitive to read the workflow when there are multiple *End* activities. In the following example, the paths to the two *Ends* are mutually exclusive execution paths. If this was a large workflow, with many activities between **Branch** and the second *End*, the value of the multiple ends becomes apparent. Tracing a **No** response from **User is invalid** to a single *End* behind 33 other activities would be significantly more difficult. There is no risk in this workflow design because there is no reason for other activities to execute if the **End** after the **Notification** activity terminates the workflow.
Figure 870: Mutually exclusive execution paths

The next example has multiple End activities in execution paths that are not mutually exclusive. A Yes response from User is valid causes the Set Values activity to finish immediately. By reaching its End activity first, this execution path cancels the Approval for Apps and the DB Task activities, which might not be the desired outcome. If the paths are all expected to complete before End, the activities should come to a Join (as in the previous example) that transitions to a single End.
Note: To add the second End, right-click to copy the original End activity and paste it onto the canvas. In most cases, a single End is the best and most reliable way to ensure that all activities expected to execute prior to workflow completion, do so successfully.

ValidateUpdateSetDependencies
The ValidateUpdateSetDependencies validator identifies all the subflows called in the current workflow and determines if any of those subflows are being edited in a different (in progress) update set.

This warning informs the user that this workflow and one or more of its dependencies are being actively modified in a way that will not deploy concurrently to another instance without additional effort.

Validation summary

- **Risk:** If a parent workflow is edited in one update set and its dependent subflow is edited in another, the two workflows might not be compatible when moved to a different instance. Making independent changes, such as to common or expected values, can make the two workflows incompatible.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** There were no Update Set dependency issues found.
- **Invalid Result:** Invalid
- **Invalid Message:** This workflow has dependent workflows that are in a different update set.
• **Suggested Action:** Modify and deploy both workflows in the same update set. If you must modify dependencies in separate update sets, use one of these methods:
  
  • Ensure that all update sets migrate concurrently.
  • Prior to deploying the main flow update set, merge the dependencies into one update set before completing that update set.

• **Publishable:** Yes
• **Runnable:** Yes
• **Related Information:** *Workflow movement with update sets* on page 3652

**Troubleshooting**

A workflow is added to an update set only when the workflow is published. This validator issues a warning when either of the following conditions exist:

• A published subflow is in a different update set than the parent workflow and that update set is In progress.
• A subflow is checked out by another user, who is working in a different update set than the current user.

**Note:** This validator does not look for update sets that have been closed. It looks only at update sets that are In progress or at the update sets of all subflows being used by the current workflow that are checked out to users who are working in a different update set.

**Example**

Following is an example of an at-risk development scenario in which two users create dependencies between workflows in different update sets.

**User A:**
1. Sets Update Set A to the current update set.
2. Checks out Workflow A.
3. Changes the return value of the *String* type in Workflow A to a *Reference/User* type.
4. Publishes Workflow A, causing an entry into Update Set A.

**User B:**
1. Sets Update Set B to the current update set.
2. Checks out Workflow B.
3. Includes Workflow A as a subflow.
4. Uses the user reference return value from Workflow A as an approval assignment.
5. Publishes Workflow B, causing an entry into Update Set B.
**Risks**

- User B moves Update Set B to a different instance that has an older version of Workflow A. The return value is not a user reference, which causes the outcome of Workflow B to be different than it was when tested in development.
- User B moves Update Set B to a new instance that does not have a version of Workflow A. Workflow B experiences a validation failure at runtime and cannot execute. A log entry is added to the workflow log of the current record.

**Possible solutions**

**Solution 1**

Migrate the parent workflow and all dependent workflows to a new instance together using the same update set.

1. Set the update set to the one you want to migrate to new instances.
2. Check out and republish the workflows that need to be included.
   - This action forces an entry into the current update set.
3. Complete the update set with all dependencies.
4. Follow standard procedures for migrating update sets to local instances.

**Solution 2**

Move dependent workflows between update sets.

1. Identify the update set containing the main workflow to be migrated.
2. Navigate to **System Update Sets > Local Update Sets**.
3. Find and select the update set that contains the dependencies to the main workflow.
4. In the **Customer Updates** related list, select the workflow version of the subflow you want to move.
5. Select the update set containing the parent workflow in the **Update set** field.
   - If this field is not on the Customer Update form, **configure the form** and add the field.
6. Click **Update**.
   - The ServiceNow ITSA Suite moves the dependent subflow to the update set selected.
7. Repeat steps 4-6 to add additional dependent subflows to the parent flow update set.

---

*ValidateUpdateSetParentDependencies*

The **ValidateUpdateSetParentDependencies** validator identifies all the workflows that call the current workflow as a subflow and determines if any of those parent workflows are being edited in a different update set that is in progress.

This warning informs the user that this workflow and one or more workflows that depend on this workflow are being actively modified in a way that will not deploy concurrently to another instance without additional effort.
Validation summary

- **Risk:** If a parent workflow is edited in one update set and its dependent subflow is edited in another, the two workflows might not be compatible when moved to a different instance. Making independent changes, such as editing common or expected values, can make the two workflows incompatible.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** There were no Update Set dependency issues found.
- **Invalid Result:** Invalid
- **Invalid Message:** This workflow has dependent workflows that are in a different update set.
- **Suggested Action:** Modify and deploy both workflows in the same update set. If you must modify dependencies in separate update sets, use one of these methods:
  - Ensure that all update sets migrate concurrently.
  - Prior to deploying the main flow update set, merge the dependencies into one update set before setting that update set to complete.

- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** Workflow movement with update sets on page 3652

Troubleshooting

A workflow is added to an update set only when the workflow is published. This validator issues a warning when either of the following conditions exist:

- A published subflow is in a different update set than the parent workflow and that update set is In progress.
- A subflow is checked out by another user, who is working in a different update set than the current user.

**Note:** This validator does not look for update sets that have been closed. It looks only at update sets that are In progress or at the update sets of all parent workflows that call the current workflow and are checked out to users who are working in a different update set.

Example

Following is an example of an at-risk development scenario in which two users create dependencies between workflows in different update sets.

**User A:**
1. Sets Update Set A to the current update set.
2. Checks out Workflow A.
3. Changes the return value of the **String** type in Workflow A to a **Reference/User** type.
4. Publishes Workflow A, causing an entry into Update Set A.

**User B:**
1. Sets Update Set B to the current update set.
2. Checks out Workflow B.
3. Includes Workflow A as a subflow.
4. Uses the user reference return value from Workflow A as an approval assignment.
5. Publishes Workflow B, causing an entry into Update Set B.

Risks

- User B moves Update Set B to a different instance that has an older version of Workflow A. The return value is not a user reference, which causes the outcome of Workflow B to be different than it was when tested in development.
- User B moves Update Set B to a new instance that does not have a version of Workflow A. Workflow B experiences a validation failure at runtime and cannot execute. A log entry is added to the workflow log of the current record.

Possible solutions

Solution 1

Migrate the parent workflow and all dependent workflows to a new instance together using the same update set.

1. Set the update set to the one you want to migrate to new instances.
2. Check out and republish the workflows that need to be included.
   - This action forces an entry into the current update set.
3. Complete the update set with all dependencies.
4. Follow standard procedures for migrating update sets to local instances.

Solution 2

Move dependent workflows between update sets.

1. Identify the update set containing the main workflow to be migrated.
2. Navigate to System Update Sets > Local Update Sets.
3. Find and select the update set that contains the dependencies to the main workflow.
4. In the Customer Updates related list, select the workflow version of the subflow you want to move.
5. Select the update set containing the parent workflow in the Update set field.
   - If this field is not on the Customer Update form, configure the form and add the field.
6. Click Update.
   - The ServiceNow ITSA Suite moves the dependent subflow to the update set selected.
7. Repeat steps 4-6 to add additional dependent subflows to the parent flow update set.

ValidateInputVarUpdateSetDependencies

The ValidateInputVarUpdateSetDependencies validator examines update sets to ensure that workflow input variables for a given workflow have not been deleted in different update sets than those currently In progress.
Validation summary

- **Risk:** Workflows and their input variables are not moved together in a single update set. The deletion of input variables is captured in a different update entry. If these two entries are not in the same update set, the workflow execution can be unstable.
- **Purpose:** Determine whether input variables that belonged to this workflow were deleted in a different update set.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** There were no Input Variable Update Set dependency issues found.
- **Invalid Result:** Invalid
- **Invalid Message:** There are input variables that have been deleted and logged in a different update set.
- **Suggested Action:** If the deletion is not intended to be separate from the workflow, ensure that both update sets are committed concurrently to the new instance, or merge both payloads into a single update set.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** Workflow movement with update sets on page 3652

Troubleshooting

Workflow input variables get individual entries in the Customer Update related list in the current user's update set. This validator reports to the user when workflow input deletions have happened in an update set other than the current user's update set.

Follow the instructions for Input variable removal on page 3656 when the validator issues this warning.

ValidateWorkflowEndStages

The ValidateWorkflowEndStages validator checks that in workflows with stages, the end activity of the workflow has a stage named Complete or Completed.

If the workflow has stages associated with it, but does not have the completed stage on the end activity, then the stage indicator will not show that the workflow completed.

Validation summary

- **Risk:** The stage indicator will not show the workflow is completed.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** Workflow end stages are valid.
- **Invalid Result:** invalid
- **Invalid Message:** End activity A found with invalid stage “S.”
- **Suggested Action:** If this is not by design, make the appropriate changes.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** Workflow activities on page 3676
**Troubleshooting**

Check the workflow for an end activity. Ensure that this activity assigns a stage named Complete or Completed.

*ValidateWorkflowStageColumn*

The *ValidateWorkflowStageColumn* validator detects and reports when the stage field (stage column) for a workflow is not correct or is unusable.

**Validation summary**

- **Risk:** The stage indicators may not display appropriate information.
- **Severity Level:** Warning
- **Valid Result:** Valid
- **Valid Message:** Workflow stage values are valid.
- **Invalid Result:** Warning
- **Info Summary:** Stage warnings found.
- **Invalid Messages:**
  - Table T does not have a column named C.
  - Workflow for table T has stages, but no stage column.
  - Table T has a stage column "C," but no stages are set by activities.
  - Workflow on table T has stage column "C" that is not type=workflow.

- **Suggested Action:** If this is not by design, make the appropriate changes.
- **Publishable:** Yes
- **Runnable:** Yes
- **Related Information:** Workflow activities on page 3676, Workflow stages

**Troubleshooting**

To check the stage column:

1. Open and check out a workflow.
2. Open the workflow version properties dialog by clicking the menu icon and selecting **Properties**.
3. View the **Stages** tab or section.
4. Check that the assigned stage column is actually a column in the table to which the workflow is associated.
5. Check that the column is type=workflow.

Tip: Stage columns should not be choice lists. If they are, the list appears read-only in form views, since changing that column value outside the workflow engine does not ensure safe tracking of stage states.

*ValidateWorkflowStateValues*

The *ValidateWorkflowStateValues* validator checks a number of stage aspects in workflow activities for correctness.

This validator has multiple possible error messages.
Validation summary

- **Risk**: The stage indicators may not display appropriate information.
- **Severity Level**: Warning
- **Valid Result**: Valid
- **Valid Message**: Workflow stage values are valid.
- **Invalid Result**: invalid
- **Warning summary**: Stage warnings found.
- **Invalid Messages**:
  - Stage with empty name is not allowed.
  - Stage with empty value is not allowed.
  - Cannot have more than one stage with the same name: x.
  - Cannot have more than one stage with the same value: y
- **Suggested Action**: If this is not by design, make the appropriate changes.
- **Publishable**: Yes
- **Runnable**: Yes
- **Related Information**: Workflow activities on page 3676

Troubleshooting

Use the following procedure to troubleshoot this validator:

1. Open and check out a workflow.
2. Open the list of stages for the workflow by clicking the menu icon and selecting **Edit Stages**.
3. Check the names and values.
   
   Ensure that the names and values are unique and are not empty.

4. If the same stage name or value appears more than once, remove one of the rows. It is then very important to go through the workflow and reassign stages in the activities that used the removed stage.
Workflow Stages

![Workflow Stages](image)

**Figure 872: Validate Workflow State Values**

Tip: Make a list of which activities assign which stages.

**Workflow concepts**

You can do many things using the workflow editor.

- Modify core *activities* and *conditions*.
- *Create custom activities* and reuse the data for other workflows.
- Download activity packs from the ServiceNow Store and create packs for upload.
- *Edit workflows* graphically.
• Define transitions between workflow activities.
• For the table that corresponds to the workflow, customize business rules.
• Summarize workflow progress through stages.
• Validate workflows to identify potential problems.
• Publish workflows for other users.
• Edit multiple tables without needing to directly modify them.

Workflow stages

Workflows can indicate workflow progress by updating any field designated as a stage field.

For example, the Incident [incident] table has an Incident state field that indicates progress, but the service catalog uses the Stage field.

To indicate the workflow's progress through the possible stage values, the interface updates the Stage field selected in the workflow properties. Available fields depend on the table used by the workflow. If the field provides a choice list, then the choices are available as stage values for the workflow. If the field is a workflow field, it displays an icon to indicate the workflow's progress, as with the Service Catalog's Stage field.

![Stage field](image)

Figure 873: Service Catalog Stage field with icons

After stages are added to the workflow, they can be assigned to each workflow activity. If an activity with an assigned stage is encountered when the workflow runs, the workflow engine assigns the stage to the record associated with the workflow context.

For workflows that use the Requested Item [sc_req_item] table, the stage field is automatically set to the Stage field of the table and cannot be changed. The stage state displayed for a workflow running on the Requested Item table is based on the state of the workflow activities. If an activity is active, then the stage is shown with the state of In progress. If an activity is in the Pending or Completed state, the stage reflects this state.

How stage values are derived

Stage values are derived from various sources in the interface.

Stage values are derived from the following sources:
• **Choices for Stage column**: Choices defined for the column selected as the **Stage** column for the workflow.

• **Default stages for table**: Stages defined in the Stage Default [wf_stage_default] table for the table selected.

• **Workflow-specific stages**: Applied only to the workflow for which they were defined in the Workflow Stage [wf_stage] table.

• **Stage values in existing records**: Values from the designated **Stage** column in the table assigned to the workflow are inherited from existing records.

If the stage field for a workflow is the table column named **Stage**, then the progress of the workflow appears in any list view containing the **Stage** column.

![Figure 874: Stages in a list](image)

Stage values shown in the list views are accompanied by the state, based on the workflow activities being executed. If an activity has a stage specified for it, and the activity is currently active in the workflow, then the stage is shown with a state of **In progress**. Similarly, if the activity is in the **Pending** or **Completed** state, the stage reflects this state.
Example

If the workflow table is Request Item [sc_req_item], then the stage field is automatically set to the Stage column of that table and cannot be changed. The following stage values for the request item are displayed in a choice list from the Dictionary Entry [sys_dictionary] table:

- Waiting for Approval
- Fulfillment
- Delivery

In addition, the Request Item table has the following default stages:

- Request Cancelled
- Completed

When you edit available activity stages in the workflow editor, the list displays the following stage values:

- Waiting for Approval
- Fulfillment
- Delivery
- Request Cancelled
- Completed

The following diagram depicts the process used to gather stage values from the Request Item table to populate the Stages list in workflow activities.
**Note:** If you are creating a workflow with a table other than Request Item [sc_req_item], you must select a **Stage field** in the workflow properties for the workflow to have stages.

**Use workflow stages**

You can add or modify workflow stages.

1. Navigate to **Workflow > Workflow Editor**.
2. Create a new workflow by clicking **New** or open an existing workflow.
3. In the Workflow Properties form, if the table is not the Requested Item [sc_req_item] table, select a field to display stages in the **Stage** field property.

*Available fields* depend on the table selected for the workflow.

4. After assigning a list of stages to the workflow, you can set a stage value in any of the workflow activities that provide a **Stage** field in their dialog box.

Setting a stage value in a workflow activity tells the workflow engine to assign that stage to the record associated with the workflow context when the activity is encountered during a run.

When an activity starts with a specified stage, the platform updates the **Stage** field with the current value. In workflows run against the Requested Item [sc_req_item] table, the `sc_req_item.stage` field is defined as a workflow type field.

When displaying the stages for a workflow on the Requested Item [sc_req_item] table, the stage state is based on the state of the workflow activities. If an activity has a stage specified for it, and the activity is currently active in the workflow, then the stage is shown with an **In progress** state. Similarly, if the activity is in the **Pending** or **Completed** state, the stage reflects this state.

5. To edit the workflow-specific stages for an existing workflow, click the gear icon in the header bar and select **Edit Stages**. Again, these stage values are combined with the choices, defaults, and existing values.

*Workflow stages example*

To optimize the use of stages, you can assign a stage to multiple activities in a workflow.
For example, if your workflow uses the following activities to create tasks:

1. Get approval
2. Order equipment
3. Receive equipment
4. Add equipment to CMDB
5. Set up equipment
6. Install software
7. Configure software
8. Deliver to user

The following stages might be used:

- Approval
- Order
- Configure
- Deliver

A good practice is to assign stages to the activities as follows:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Assigned stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get approval</td>
<td>Approval</td>
</tr>
<tr>
<td>Order equipment</td>
<td>Order</td>
</tr>
<tr>
<td>Receive equipment</td>
<td>Order</td>
</tr>
<tr>
<td>Add equipment to CMDB</td>
<td>Order</td>
</tr>
<tr>
<td>Set up equipment</td>
<td>Configure</td>
</tr>
<tr>
<td>Install software</td>
<td>Configure</td>
</tr>
<tr>
<td>Configure software</td>
<td>Configure</td>
</tr>
<tr>
<td>Deliver to user</td>
<td>Deliver</td>
</tr>
</tbody>
</table>

When the Order equipment, Receive equipment, or Add equipment to CMDB activity is active, the Stage display shows that the Order stage is In progress.

Add default stages for a table

You can assign a stage set as a default set to any number of tables.

The stages in the assigned stage set pre-fill the workflow version stages when you create a new workflow for the associated table.

1. Navigate to Workflow > Default Stages (by table).
2. Click New.
3. Select a stage Set.
   - You can create a new set by clicking New, typing a name, and clicking Submit.
4. Select a corresponding Table.
5. Click **Submit**.
   The stages in the selected stage set are automatically added to any new workflows that use the corresponding table.

Add a stage to an existing workflow

If a stage required for a workflow has not been imported or is not in the stage set assigned to the workflow table, you can add it to the workflow manually.

1. Navigate to **Workflow > Workflow Editor**.
2. Open and check out the workflow.
3. In the title bar, click the menu icon and select **Edit Stages**.
4. In the Workflow Stages list, click **New**.
5. On the Workflow Stage form, fill in the fields as appropriate.
   Do not use a **Name** or **Value** field value that already exists in the base system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the stage as it appears in workflow fields.</td>
</tr>
<tr>
<td>Value</td>
<td>The value of the stage when it is referenced from elsewhere in the system, such as in a script.</td>
</tr>
<tr>
<td>Duration</td>
<td>The default duration for the stage. Currently not used.</td>
</tr>
</tbody>
</table>

6. Click **Submit**.
7. Publish the workflow.

Add a stage to a workflow activity

After stages are added to a workflow, you can assign them to the workflow activities.

1. Navigate to **Workflow > Workflow Editor**.
2. Open and check out the workflow.
3. Double-click the activity.
4. In the **Stage** field on the Activity Properties form, find and select the stage to display when the workflow reaches the activity.
5. Click **Update**.

Translate workflow stages

You can provide translations for workflow stage names so the names appear in the language selected for the instance.

The instance must already have **Internationalization support** on page 965 installed.

Language-specific text appears in **workflow fields** displayed in a list, **workflow stage sets**, and the workflow editor for users with that language selected. Language-specific text does not automatically appear when displaying stages on a form. To translate stages on a form, add translated text to the workflow field choices.
You can add translated text for any language enabled on the instance. For example, to translate workflow stages into French:

1. Set the interface language to French.
2. Navigate to Workflow > Workflow Editor.
3. Open and check out a workflow.
4. In the title bar, click the menu icon and select Edit Stages.
5. Edit the Name field for each stage and enter the text to display to French users. Do not change the Value field.
6. Confirm that the workflow stages display the French text.

Alternatively, to translate stages for multiple workflows at once, you can directly edit the Workflow Stages [wf_stage] table. For example, to translate workflow stages from multiple workflows into French:

1. Set the interface language to French.
2. In the application navigation filter, enter wf_stage.list.
3. Edit the Name field for each stage and enter the text to display to French users. Do not change the Value field.

Workflow stage sets

You can create stage sets that are named groups of stages commonly used together. These stage sets can be used in any number of workflows on different tables.

A single stage set usually represents a process, such as the stages required to display the progress of a service catalog request. You can also assign a stage set as the default set for workflows on specific tables.

In an active context, workflow stages provide summary-level feedback about the progress of a workflow. Stage icons display the status of each activity as it is being driven by a workflow. The system updates a Stage field, defined in the workflow properties, to indicate the workflow's progress through the possible stage values.

Create a new stage set

You can create a new stage set by creating a stage set record and adding stage set entries manually. Create a stage set record only when you want additional stages that are not available on the table by default.

1. Navigate to Workflow > Stage Sets.
2. Click New.
3. Enter a Name that indicates the purpose of the stage set.
   For example, you can create a Requested Item stage set to hold the stages commonly used by inventory tracking workflows or service catalog fulfillment workflows.
4. Click Submit.
5. Open the new stage set record.
6. In the Stage Set Entries related list, click New.
   Each stage set entry can be used as the Stage for an activity in a workflow that uses this stage set.
7. Enter a Name that indicates the stage name to appear in workflow fields.
8. Enter a Value to use when referencing the entry, such as in a script.
9. Click Submit.
10. In the Stage Set Entries related list, ensure that each entry has a unique Order value.
    Enter a low value for stages that should appear early in the workflow and a higher value for later stages.
The stage set can be added to a workflow or assigned as the default stage set for workflows that are created for a specific table.

- *Add a stage set to a workflow* on page 3624
- *Add default stages for a table* on page 3620

*Import stages from a choice list*

You can import the choice list values of a workflow field as stages for a workflow.

Create a choice list as follows:

- Add a custom field of Type *Workflow* to the table.
- Configure the custom field to use a choice list.
- If you are creating a new field, set the *Choice List type* to *Display without --None--* and create the choices for the newly created field.

For more information, see *Create a workflow stage field* on page 3625.

1. Navigate to *Workflow > Workflow Editor*.
2. Open and check out the workflow.
3. In the title bar, click the menu icon and select *Properties*.
4. In the Workflow Properties dialog box, click the *Stages* tab.
5. From the **Stage field** list, verify that the correct workflow field is selected.
6. In the Related Links section, click **Import Stages from Choice List**.
7. In the dialog box asking you to confirm that you want to import the choice list, click **OK**.
8. Click **Update**.

The stage set can be added to a workflow or assigned as the default set for workflows that are created for a specific table.

- [Add a stage set to a workflow](#) on page 3624
- [Add default stages for a table](#) on page 3620

**Export a stage set from a workflow**

You can create a new stage set by exporting the stages from an existing workflow as a set, instead of manually adding stage set entries to a stage set record.

1. Navigate to **Workflow > Workflow Editor**.
2. Open and check out the workflow containing stages that you want to export as a new stage set.
3. In the title bar, click the menu icon and select **Edit Stages** to open the Workflow Stages dialog box.
4. In the Related Links section, click **Export to Stage Set**.
5. Enter a unique **Name** for the new stage set.
6. Click **OK**.

The stage set can be added to a workflow or assigned as the default set for workflows that are created for a specific table.

- [Add a stage set to a workflow](#) on page 3624
- [Add default stages for a table](#) on page 3620

**Add a stage set to a workflow**

You can add any number of stage sets to an existing workflow.

When multiple stage sets have stage set entries with the same **Value**, the stage appears on the workflow only once.

1. Navigate to **Workflow > Workflow Editor**.
2. Open and check out the workflow.
3. In the title bar, click the menu icon and select **Edit Stages** to open the Workflow Stages dialog box.
4. In the Related Links section, click **Import from Stage Set**.
5. Select the stage set.
6. Click **Ok**.
7. Import additional stage sets as needed for the workflow.

After you add all necessary stage sets to the workflow, you can add them to the workflow activities. For more information, see [Add a stage to a workflow activity](#) on page 3621.

**Use a default stage set**

You can assign a stage set as a default set to any number of tables.

The stages in the assigned stage set pre-fill the workflow version stages when you create a new workflow for the associated table.

1. Navigate to **Workflow > Default Stages (by table)**.
2. Click **New**.
3. Select a **Table**.
4. Select the **Set** you want to assign to the selected table.
5. Click **Submit**.

Create a workflow stage field

Workflows can provide a summary of workflow progress by updating any field of the **Workflow** type. If the field is a workflow field, it displays an icon to indicate the workflow stage progress.

Role required: admin

In a form, the workflow field displays the current stage as a choice list value.

In a list, the workflow field displays stage icons that represent the series of states, stages, phases, or tasks within a workflow. The stage icons are populated using the activity stages from the associated workflow. For more information, see *Workflow stages* on page 3587.

To create a workflow stage field:

1. Navigate to **Workflow > Default Stages (by table)**.
2. Right-click near a column header and select **Configure > List Layout**.
3. Create a new field in lower right.
4. On the form view of the table used by the workflow, create a field with the **Type** set to **Workflow**.
   For detailed steps, see *Add a new field to a table* on page 783.
5. Click **Save**.
6. Right-click the field label and select **Configure Dictionary**.
7. Open a dictionary entry.
8. In the Choice List Specification section of the Dictionary Entries form, select **Dropdown with --None--** from the **Choice** list.
9. Click **Update**.
10. To ensure that the workflow field properly displays stages on the form, define a choice for each workflow stage.
    - To create workflow stages using a choice list that already has choices, you can import the stages from the choice list. A stage is created for each choice defined for the **Stage field** selected in the workflow properties. For detailed steps about importing a choice list, see *Import stages from a choice list* on page 3623.
    - The **Value** of each choice must match the **Value** of the corresponding workflow stage.
• Do not use spaces in the **Value** field for either choices or stages. Use underscores in place of spaces.
• Make the choice list read-only. If a user changes the stage value for a record from the choice list rather than allowing the workflow to control the value, the workflow-driven, legacy, and linear renders may not work as intended. You can also use business rules or events to ensure the workflow progresses accordingly.

For detailed steps on defining choice lists, see *Define an option for a choice list* on page 843.

11. To see workflow stages as icons, add the field to the list layout of the workflow table.
   For detailed steps, see *Configure the list layout* on page 672.

After you create or update the workflow field:

- Unless the workflow uses the Requested Item [sc_req_item] table, ensure that the workflow field is selected from the **Stage field** list in the workflow properties. For detailed steps, see *Select a stage field* on page 3626. If a workflow uses the Requested Item table, the stage field is automatically set to the **Stage** field of the table and cannot be changed.
- Work through the workflow. Any other updates made to the workflow field, such as updates from business rules or other scripts, can interfere with displaying workflow stages.

**Select a stage field**

A **Stage field** allows the workflow context to show additional workflow information, such as the stage name and the estimated completion time for an activity.

Ensure that the workflow field you want to use as the stage field is configured to properly display stages. For detailed steps, see *Create a workflow stage field* on page 3625.

Unless the workflow uses the Requested Item [sc_req_item] table, you can specify which field from the workflow table is the stage field. For workflows that use the Requested Item [sc_req_item] table, the stage field is automatically set to the **Stage** field of the table and cannot be changed.

To add or edit a workflow stage field:

1. Navigate to **Workflow > Workflow Editor**.
2. Create or check out the workflow.
3. In the title bar, click the menu icon and select **Properties**.
4. In the Workflow Properties dialog box, click the **General** tab.
5. In the **Table** list, verify that the table containing the workflow field is selected.
6. Click the **Stages** tab.
7. From the **Stage field** list, select the workflow field.
8. Click **Update**.

**Workflow field icons**

A workflow stage field displays icons to indicate the workflow stage.

Based on the stage renderer selected for the workflow, these icons may display a tooltip with additional information.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Workflow stage</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Current active step</td>
</tr>
</tbody>
</table>
Stage tooltips
Based on the stage renderer selected for a workflow, workflow stage icons may display tooltips with detailed information about a stage.

Table 1245: Stage renderers and tooltip behavior

<table>
<thead>
<tr>
<th>Renderer</th>
<th>Tooltip behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy</td>
<td>When a user points to a <strong>Waiting for Approval</strong> stage, the tooltip displays the name of the approver. A tooltip does not appear when a user points to other stage icons.</td>
</tr>
</tbody>
</table>
If you do not want the approver’s name included in the tooltip, navigate to System Properties > Service Catalog and clear the **Show the current pending approver’s name in the stage widget mouseover** check box.

![Figure 875: Property for displaying pending approver’s name](image)

**Workflow stage renderers**

Workflow stage renderers determine how a workflow displays stages in a workflow field.

There are multiple renderers available.

<table>
<thead>
<tr>
<th>Renderer</th>
<th>Tooltip behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other renderers</td>
<td>When a user expands the stage field, the activity state is displayed in parentheses next to the stage, for example, <strong>In progress</strong> or <strong>Pending - has not yet started</strong>.</td>
</tr>
</tbody>
</table>

**Note:** Most workflows should use the workflow-driven renderer. This renderer is used by default for all workflows. Use a different stage renderer only after careful consideration and to satisfy specific requirements for how the stages appear.

**Workflow-driven**

Use the workflow-driven renderer as much as possible. This renderer displays icons for stages using the stage state controlled by the workflow. Icons are displayed in a way that is meaningful for many situations. This renderer can display stages from a main workflow and subflows. The order of the icons is determined by the expected path of the executing workflow. As the workflow progresses, stages on paths that the
workflow did not take are removed from the display. Stages from paths other than the expected path are not included unless the workflow follows that path.

The **Stage order** field on the **Stages** tab has two options:

- Computed uses the actual workflow path in order
- User-specified uses the order specified in the **Order** column on the Workflow Stages record

**Main flow**

The main flow renderer displays icons for stages defined in the main workflow only. Use this renderer when you do not want to expose the details of the subflows. For example, a single main workflow may run several subflows to handle implementation details. The stages in these subflows do not provide useful information for the user who starts the workflow, but are useful when editing the subflow. In this scenario, using the main flow renderer leads to the best user experience. The stage field displays the high-level process of the workflow without exposing unnecessary details.

The **Stage order** field on the **Stages** tab has two options:

- Computed uses the actual workflow path in order
- User-specified uses the order specified in the **Order** column on the Workflow Stages record

**Linear**

Linear rendering displays all stage icons from the main workflow and all subflows within a single workflow field on a list or form. It displays icons in a linear sequence regardless of the paths the workflow follows as it executes. This renderer uses stages defined in both the main workflow and any subflows that the main flow launches. The icons appear in the user-specified order. Skipped stages do not appear.

Use this renderer when the workflow stages must display in a consistent order and the actual details of how the workflow runs are less important. For example, a workflow may revisit or revert to previous stages based on one or more **Condition workflow activities** on page 3703. Displaying these loops in the stage field does not provide useful information to the user. In this scenario, using the linear renderer leads to the best user experience. The stage field displays the predefined process, including stages from subflows, in a consistent order.

For example, you can create a service catalog workflow that uses a subflow to complete delivery of the requested item. The following images illustrate the main flow and subflow for ordering corporate-branded supplies.
Figure 876: Linear main flow

Figure 877: Linear subflow

The linear renderer displays all these stages in one workflow field. The parent workflow specifies the Waiting for approval, Fulfillment, and Completed stages. The subflow specifies the Delivery and Waiting for Task, and Delivery OK stages.

Progress bar

The progress bar renderer displays a single progress bar instead of a sequence of stage icons. This renderer is particularly useful when you want to display the general progress of the workflow as a percent. Each stage in the workflow controls an equal percentage of the progress bar. For example, if a workflow has 10 stages, reaching each stage causes the workflow field to display an additional 10% of the progress bar. Stage names do not appear in the progress bar.
The progress bar renderer provides a way to display the status of a workflow that has a large number of stages. Other rendering options may provide a better user experience when rendering fewer than four stages in a single workflow.

The **Stage order** field on the **Stages** tab has two options:

- Computed uses the actual workflow path in order
- User-specified uses the order specified in the **Order** column on the Workflow Stages record

Figure 878: Workflow field progress bar

**Legacy**

The legacy renderer displays stages in the same way as releases prior to Dublin. When an instance is upgraded from a release prior to Dublin, all existing workflows are set to use the legacy renderer. Whenever possible, use a different stage renderer instead. If you want your workflow to maintain pre-Dublin behavior, use the Legacy option. The legacy renderer sometimes assigns incorrect data to and/or reports incorrect data from the workflow stage field. If possible, we recommend all workflows use the Workflow-driven renderer.

With the legacy renderer, you can create and reference a custom workflow field icon set by setting the icons attribute to a new script include. For example, to use the `WorkflowIconsSCR` script include to define which icons to use, add the attribute `icons=WorkflowIconsSCR` to the **Attributes** field of the dictionary entry for the workflow field. To use the default icon display behavior, use the attribute `icons=WorkflowIconsStages`.

The legacy renderer works with all tables except the Requested item [sc_req_item] table. In this case, use the requested item renderer instead.

**Requested item**

The requested item renderer functions the same way as the legacy renderer, but is for use with the Requested item [sc_req_item] table.

Select a stage renderer

In most cases, the default workflow-driven renderer should be used. If you have specific requirements for displaying stages, you can select a different stage renderer.

Consider the following when selecting a stage renderer:

- Use the workflow-driven renderer if possible. This is the default renderer that should be used in most cases.
- Use the legacy renderer only if your instance was upgraded from a release prior to Dublin and you want your workflow to maintain pre-Dublin behavior.

To use linear, main flow, or progress bar rendering, satisfy the following requirements.
Table 1246: Linear and progress bar renderer requirements

<table>
<thead>
<tr>
<th>Renderer</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>• The parent workflow and all subflows must modify the same current record.</td>
</tr>
<tr>
<td></td>
<td>• In the properties for all subflows, the <strong>Stage field</strong> value for all subflows must match that of the parent workflow.</td>
</tr>
<tr>
<td></td>
<td>• The parent flow and all subflows must contain the same stages. Use a stage set to ensure the parent flow and all subflows have the same stages.</td>
</tr>
<tr>
<td></td>
<td>• On the workflow canvas, each workflow needs only the stages used directly by that workflow. It is not necessary to add stages from subflows to activities on the parent flow, or stages from the parent flow to activities on the subflows.</td>
</tr>
<tr>
<td>Main flow</td>
<td>• The workflow contains subflows.</td>
</tr>
<tr>
<td></td>
<td>• You do not want or need to reveal the details about the subflows.</td>
</tr>
<tr>
<td>Progress bar</td>
<td>• The workflow properties must have a <strong>Stage ordering</strong> value of <strong>User-defined</strong></td>
</tr>
<tr>
<td></td>
<td>• There must be workflow stages within the workflow.</td>
</tr>
<tr>
<td></td>
<td>• Because stage names do not appear in a workflow field when using the progress bar renderer, you can simplify stage names to represent a percentage of the workflow. For example, if a process has four main steps, name the workflow stages as 25%, 50%, 75%, and 100%. Enter the numerical value of each stage, such as 25, in the <strong>Value</strong> and <strong>Order</strong> fields.</td>
</tr>
</tbody>
</table>

To select a stage renderer:

1. Navigate to **Workflow > Workflow Editor**.
2. Open and check out the workflow.
3. In the title bar, click the menu icon and select **Properties**.
4. In the Workflow Properties dialog box, click the **Stages** tab.
5. From the **Stage rendering** list, select a stage renderer.
   - If you are using two workflows to update two unique workflow fields on a single record, both workflows must use a non-legacy renderer. You can select a different stage renderer for each workflow but do not select **Legacy** for either one.
   - If you want to use the linear renderer, make sure you select **Linear** in the properties for the parent workflow and all subflows.
6. Click **Update**.

**Workflow fields with deleted records**

Workflow fields may indicate when a record required by the workflow is deleted.

After a referenced record is deleted, the reference in the primary record is no longer valid. If the stage renderer detects a reference that is no longer valid, the stage field displays a message about the deleted record.

Administrators can restore deleted records. For more information, see *Restore a deleted record and reference* on page 1626.

![Workflow fields with deleted records](image)

**Figure 879: Workflows with missing records**

The previous image shows a list with several workflows. The top request has an associated request item, but the item does not have an associated workflow context. The bottom request does not have an associated request item.

**Workflow versions**

To prevent users from making changes to a workflow that affect other users of the system, workflows must be checked out before they can be edited.

Only one user can check out a workflow at a time. When a workflow is checked out, changes apply only to the user who has the workflow checked out. Other users can continue to use the published workflow. After the changes are complete, the workflow can be published so that it is available to all users.

---

**Note:** Because each workflow has a unique sys_id, different workflows can have the same name. This is typically expected in a domain-separated environment where users in different companies cannot see each other's workflows because they are in different domains. However, this can lead to confusion in other environments. In general, give each workflow a unique name to prevent workflow designers from making changes to the wrong workflow.

When a new version of an existing workflow is published, the changes are not applied to running workflow contexts. Any currently running workflow context continues using the workflow version that was available when the workflow started. The next time the workflow runs, it uses the updated, published version.
Workflow scope

Workflow application scope determines the access that an application has to the information in a workflow, specifically to the data contained in the activities in that workflow.

When a workflow is created, it inherits the application scope from the gear menu for the logged in user. This scope cannot be changed in the workflow editor. When the workflow executes, it runs in this scope and can only be called from a different application if the workflow’s accessibility setting permits access to all scopes (public). Otherwise, the workflow’s application scope is private to the application.

**Note:** Any script that is created in the workflow editor, such as an advanced script in an If activity, runs in the scope of the workflow. All core activities provided in the base system or for Orchestration run in the scope of the workflow.

Custom activities run in their own scope, even if it is different from that of the workflow. The scope of a custom activity can be private or public. Any script that runs inside a custom activity with a scope can only access outside artifacts that are within the scope of that activity or artifacts that are configured to run in any scope. Conversely, an outside artifact can only access the script inside that private activity if the outside artifact is running in the same scope. Activities with public scopes can interact with outside artifacts in any application scope.

You can use private activities as part of a workflow that has a public application scope. These activities are protected from reaching outside of the workflow or from being reached from outside the workflow. For details about setting application scoping for custom activities, see the field description table for the appropriate activity template.

**Note:** Custom activities uploaded to the ServiceNow Store must be configured as accessible to all application scopes.

Workflow scope restrictions

There are some restrictions to public and private application scopes.

During runtime, publicly scoped workflows can access other application resources, as long as these resources are set to be accessible to all application scopes. Privately scoped workflows in a private application scope can only access resources private to its scope. Due to scope access boundaries, any privately scoped workflows that make calls out to other scoped resources fail with either an exception or a hung activity while waiting for returned results. This occurs when making calls to these common global resources:

- ECC queues
- Tasks
- Approvals
- Events
- SLA timers
- Timers
- Script includes
- Business rules
- Workflow APIs

As you design workflows, validate the visibility and accessibility of all resources prior to deployment. See **Application scope**.

For information on how to configure the scope for a workflow, see **Workflow properties** on page 3544.
Workflows and domain separation

When domain separation is enabled, workflows and workflow activities inherit the domain of the user who publishes or creates them.

While workflows are managed by multiple tables, only the following tables are used for domain separation features:

- **Workflow [wf_workflow]**: used for delegated administration or process separation.
- **Workflow Version [wf_workflow_version]**: used for delegated administration or process separation.
- **Workflow Context [wf_context]**: used for data separation.

**Note:** The Workflow Version table [wf_workflow_version] table does not contain a domain field; Workflow Version records inherit their domain from the parent Workflow record.

The workflow editor displays a workflow's domain in the title bar after the workflow name.

---

**Figure 880: Workflow editor domain**

---

Workflows and delegated administration

Delegated administration allows child domains to inherit workflows from higher up the domain hierarchy and to override them with domain-specific versions if necessary.
Workflow records in the Workflow [wf_workflow] and Workflow Version [wf_workflow_version] tables are considered processes. A user in a child domain may check out but not copy a workflow from a parent domain. When a user in a child domain checks out a workflow from a parent domain, the system creates a version of the workflow in that user's domain. This new version is a unique record in the Workflow [wf_workflow] table. After the user publishes this new workflow, other users in the child domain use the
new workflow, which overrides the workflow from the parent domain. The original workflow in the parent domain is no longer visible to users in the child domain.

For example, a managed service provider (MSP) hosts ITSM services for several companies, including ACME and Initech, on a single instance. As administrators, the MSP creates a Service Catalog Request workflow that applies to all domains because it was created in the TOP domain, which is the highest domain in the domain hierarchy. This workflow overrides the global Service Catalog Request workflow and specifies that all Service Catalog requests over $750 require approval. Because of delegated administration, every domain in the hierarchy sees and uses this workflow. Now suppose the ACME domain requires a different approval policy where requests over $500 require approval. The MSP creates another version of the Service Catalog Request workflow in the ACME domain. This workflow overrides the version in the TOP domain and only applies to users in the ACME domain.

Workflows and data separation

Data separation restricts workflow contexts to users who are either in the same domain of the workflow or are members of a parent domain.
Figure 882: Workflow and data separation

Workflow records in the Workflow Contexts [wf_contexts] table are considered data. Data separation restricts workflow contexts to users who are either in the same domain of the workflow or are members of a parent domain. While a user in a parent domain can see running workflows in a child domain, child domains cannot see data upwards.
in a child domain cannot see running workflows in a parent domain. If necessary, administrators can use visibility or contains domains to expand who can see domain-specific data.

For example, when an ACME user requests something from the service catalog, a Service Catalog Request workflow context is created in the ACME domain. Similarly, a service catalog request from an Initech user creates a workflow context in the Initech domain. An MSP user in the TOP domain can see both workflow contexts because it is the parent domain for both the ACME and Initech domains. However when an ACME or Initech user logs in, data separation prevents them from seeing each other's service catalog requests. This is expected behavior because each workflow context contains data specific to that domain, such as the item requested and the request's approval history.

Workflow permissions

When a user starts a new workflow, the workflow runs with that user's domain and credentials.

The workflow preserves a user's domain and credentials until an activity causes the workflow to wait, such as an approval activity waiting for approval or rejection. When the stopped workflow resumes, such as when a user approves a request, the workflow uses the credentials of the approving user, but continues to run within the domain of the original user.

Workflow engine operation order

The workflow engine runs in a predefined order relative to business rules and database operations.

The Run after bus. rules run workflow property defines if a workflow is Default or Deferred.

The diagram below shows the workflow engine order of operations and when Default and Deferred flows are executed. For a more general overview of engine operation order, see Execution order of scripts and engines on page 470.
Before business rules (order <1000)
Do not use current.update() here

Default workflows
Do not use current.update()

Before business rules (order >=1000)
Do not use current.update()

Database operation (insert, update, delete)
The current record is inserted or updated at the end of the database operation.

After business rules (order <1000)
Use current.update() if there is no deferred workflow

Deferred workflows
Do not use current.update()

Always use current.update().
Workflow caching

The workflow engine caches commonly-used published workflows to improve performance.

Caching significantly reduces the number of database queries per workflow. By default, the engine caches up to 300 unique workflow versions. Caching very large workflows may reduce this number as the cache size cannot exceed the Java Virtual Machine (JVM) heap size.

To change the maximum number of cached workflow versions, navigate to Workflow > Administration > Properties and modify the value of the The max number of models that will be concurrently held in the LRU cache (glide.workflow.model.cache.max) property. You must restart the instance to apply this change.

Workflow tables

For full flexibility, workflows store information over a number of different tables.

Usually tables containing workflow information are not edited one-by-one. Instead, use the Workflow Editor to edit workflows. The following lists are provided for reference purposes.

Table 1247: Workflow tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflows</td>
<td></td>
</tr>
<tr>
<td>Column Renderer [column_renderer]</td>
<td>A renderer widget for a stage column. Stage renderers are written in Jelly as a UI Macro. The default is Workflow-Driven; it covers most workflow related stage scenarios.</td>
</tr>
<tr>
<td>Version [wf_versionable]</td>
<td>Tracks different versions of element definitions [wf_element_activity].</td>
</tr>
<tr>
<td>Workflow [wf_workflow]</td>
<td>The master records of workflows.</td>
</tr>
<tr>
<td>Workflow Context [wf_context]</td>
<td>Individual instances of a workflow being used.</td>
</tr>
<tr>
<td>Workflow Execution [wf_workflow_execution]</td>
<td>Synthetic “current” records for workflows that run on Global.</td>
</tr>
<tr>
<td>Workflow Instance [wf_workflow_instance]</td>
<td>Connections of workflows to subflows.</td>
</tr>
<tr>
<td>Workflow Version [wf_workflow_version]</td>
<td>Particular versions of a workflow, either published versions or versions that have been checked out.</td>
</tr>
<tr>
<td>Activities</td>
<td></td>
</tr>
<tr>
<td>Activity Variables [wf_activity_variable]</td>
<td>Variables for activities.</td>
</tr>
<tr>
<td>Workflow Activity [wf_activity]</td>
<td>Activities as they are being used in workflows.</td>
</tr>
<tr>
<td>Workflow Activity Definition [wf_activity_definition]</td>
<td>Definitions of activities that can be used in a workflow.</td>
</tr>
<tr>
<td>Workflow Executing Activity [wf_executing]</td>
<td>Individual instances of activities being performed in active contexts.</td>
</tr>
</tbody>
</table>

Workflow components
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Provider [wf_element_provider]</td>
<td>Template definitions for custom activities.</td>
</tr>
<tr>
<td>Variable [item_option_new]</td>
<td></td>
</tr>
<tr>
<td>Workflow Condition [wf_condition]</td>
<td>All of the defined conditions in workflows.</td>
</tr>
<tr>
<td>Workflow Queued Command [wf_command]</td>
<td>Temporary internal storage for workflows that are currently executing.</td>
</tr>
<tr>
<td>Workflow SC Variable [wf_variable]</td>
<td>The Service Catalog variables for a workflow.</td>
</tr>
<tr>
<td>Workflow Schedule [wf_workflow_schedule]</td>
<td>Definitions of the times to run specific workflows.</td>
</tr>
<tr>
<td>Workflow Transition [wf_transition]</td>
<td>All of the defined transitions in workflows.</td>
</tr>
<tr>
<td>History</td>
<td></td>
</tr>
<tr>
<td>Workflow Activity History [wf_history]</td>
<td>The history of executed activities.</td>
</tr>
<tr>
<td>Workflow Log Entry [wf_log]</td>
<td>All of the events and history of the workflow.</td>
</tr>
<tr>
<td>Workflow Transition History [wf_transition_history]</td>
<td>The history of executed transitions.</td>
</tr>
<tr>
<td>Stages</td>
<td></td>
</tr>
<tr>
<td>Stage Default [wf_stage_default]</td>
<td>Definitions of default stage fields for tables to use.</td>
</tr>
<tr>
<td>Stage Set [stage_set]</td>
<td>A named set of stages that can be used to populate workflow stages for multiple workflows.</td>
</tr>
<tr>
<td>Stage Set Entry [stage_set_entry]</td>
<td>The stages that belong to a named stage set.</td>
</tr>
<tr>
<td>Stage Set for Table [stage_set_table]</td>
<td>Defines a relationship of a stage set to a table so that the stage set can be used as the default stages when a new workflow is created for the table. This replaces the wf_default_stage table and is the view that shows when you click Default Stages (by table) in the menu.</td>
</tr>
<tr>
<td>Workflow Stage [wf_stage]</td>
<td>Definitions of stages used by workflows.</td>
</tr>
</tbody>
</table>

**Workflow administration**

Tailor workflows exactly the way you want them.

**Workflow roles**

Certain roles are required to use workflows.
Table 1248: Workflow roles

<table>
<thead>
<tr>
<th>Role title [name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity creator [activity_creator]</td>
<td>Creates and edits custom workflow activities, reuses custom activity data, and manages activity packs downloaded from the ServiceNow Store.</td>
</tr>
<tr>
<td>Web service administrator [web_service_admin]</td>
<td>Accesses and uses REST and SOAP messages in the Orchestration activity designer. Creates and edits custom activities that use the REST web service and SOAP web service templates.</td>
</tr>
<tr>
<td>Workflow administrator [workflow_admin]</td>
<td>Creates, edits, publishes, and deletes graphical workflows.</td>
</tr>
<tr>
<td>Workflow creator [workflow_creator]</td>
<td>Creates new graphical workflows.</td>
</tr>
<tr>
<td>Workflow publisher [workflow_publisher]</td>
<td>Publishes graphical workflows.</td>
</tr>
</tbody>
</table>

Administering workflow contexts

The workflow context performs the activities and transitions defined in the workflow with the new record as current.

Workflow in ServiceNow names a running workflow a Workflow Context. The Workflow Context maintains the state of the overall process in the Workflow Context record. The Workflow Context maintains the state of the individual activities as they execute in a series of related lists. These lists maintain the state of currently executing activities, the result of finished activities, and the execution path the workflow took through the process model.

The Workflow Context canvas provides a visual representation of the execution path the workflow took through the process model. The state of each activity (finished, executing, cancelled, error) is represented using the color palette. The executed paths are represented in the color blue; the non-executed paths are represented in grey. Active and historic workflow contexts, as well as the activities within them, can be viewed using the Live Workflows section of the Workflow application menu.

Viewing a workflow context

Workflow contexts can be found in two places:

- From the Workflow Context related link on the form of the task being powered by the workflow.
- By navigating to Workflow > All Contexts and selecting an active context.

Displaying workflow progress

Two related links on the Workflow Context form allow you to view the progress of a workflow in different formats.
- **Show Timeline** displays the workflow context as a *timeline*.
- **Show Workflow** displays the workflow context in the graphical workflow editor.

**Graphical interface**

To view the workflow context in the graphical workflow editor interface, click the **Show Workflow** link from either the workflow context record or the current record.

![Graphical interface of workflow](image)

**Figure 884: Show workflow**

In the top right hand corner are two controls:
<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Refresh" /></td>
<td>Refreshes the workflow context.</td>
</tr>
<tr>
<td><img src="image" alt="Question Mark" /></td>
<td>Displays a key of the colors used in the workflow to denote the state of activities and transitions:</td>
</tr>
</tbody>
</table>

### Execution order

Tooltip text in the workflow context graphical view displays the execution order of individual activities. This can be useful when troubleshooting.

1. Navigate to Workflow > Live Workflows > Active Contexts or All Contexts.
2. Open the context you want to examine.
3. Click **Show Workflow**.
4. Point to a finished or executing activity.
   A tooltip appears, showing error data, execution time, and the order in which the activity executed in the workflow. Use this data to help troubleshoot activities in an error state.
Canceling a workflow

Canceling a workflow stops the workflow from executing and sets the workflow context State to Canceled. Canceling a workflow attempts to stop the workflow gracefully by injecting a cancel command into the workflow engine.

To cancel an active workflow:

1. Navigate to Workflow > Active Contexts.
2. Select a workflow context record.
3. Select the Cancel related link.
A confirmation window appears.

4. Click OK.
   The Wait for Cancel window appears. The workflow engine attempts to cancel the workflow gracefully.

   If the workflow does not respond to the cancel command, the Force Cancel window appears.
5. Click **Force cancel** to interrupt the thread the workflow is actively executing or click **Continue waiting** to continue waiting for the workflow to cancel gracefully.

**Warning:** Whenever possible, allow a workflow to cancel gracefully. Forcing a workflow to cancel can leave related workflows and scripts in an unresolved state. You can use an on-cancel script to clean up unresolved artifacts from a cancelled workflow.

---

**Cancel a workflow with the `cancelContext(context)` script**

To cancel an executing workflow, you can use the `cancelContext(context)` script. This script can be useful in cases where a workflow must be canceled in response to an event or where a user must manually cancel a workflow.

See the ServiceNow Developers site for API information.

**Defining an on-cancel script**

Canceling a workflow can leave records or scripts in an unresolved state. For example, canceling a service catalog workflow may leave catalog items in the requesting user's cart. An administrator can specify an On-cancel script that runs when the workflow transitions to the Canceled state. This script can notify users, log information, or resolve the state of any scripts run within a workflow activity. The `sys_id` of the workflow context is available in this script using the `context_sys_id` variable.

To define this script:

1. Navigate to **Workflow > Workflow Versions**.
2. Select a workflow version that you have checked out. Workflow versions that are not checked out are not editable.
3. Edit the **On-cancel script** field. You may need to **configure the form** to add this field.
4. Click **Update**.

This example script adds a comment to a Requested Item `[sc_req_item]` record indicating the workflow for that request has been canceled.

```javascript
var grContext = new GlideRecord("wf_context");
grContext.get(context_sys_id);
var grReq = new GlideRecord("sc_req_item");
```
Scheduling a workflow

In addition to being run based on conditions, workflows can also be scheduled to perform at a particular time, similar to a scheduled job.

Scheduled workflows do not have a defined current record and do not run on specific records within a table. Due to this behavior, certain activities that depend on a current record, such as Create Task or Catalog Task are unavailable on scheduled workflows. If your workflow requires one of these activities, consider using a scheduled job that inserts a record to start the workflow instead of using a scheduled workflow.

To schedule a workflow, navigate to Workflow > Scheduled Workflows and click New. Populate the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name for the scheduled workflow.</td>
</tr>
<tr>
<td>Workflow</td>
<td>Select an existing published workflow to be triggered at the specified date, time, or interval.</td>
</tr>
<tr>
<td>Active</td>
<td>If selected, the scheduled workflow will be triggered at the appropriate time.</td>
</tr>
<tr>
<td>Application</td>
<td>Specifies the type of application, such as Global.</td>
</tr>
<tr>
<td>Run</td>
<td>A choice list to determine when the workflow should be triggered. Options are:</td>
</tr>
<tr>
<td></td>
<td>- Daily: At a particular hour every day.</td>
</tr>
<tr>
<td></td>
<td>- Weekly: On a particular day of the week.</td>
</tr>
<tr>
<td></td>
<td>- Monthly: On a particular day of the month</td>
</tr>
<tr>
<td></td>
<td>- Periodically: After every set duration.</td>
</tr>
<tr>
<td></td>
<td>- Once: At one specific date and time.</td>
</tr>
<tr>
<td>Time</td>
<td>For daily/weekly/monthly scheduled workflows, the time of day to run the workflow.</td>
</tr>
<tr>
<td>Day</td>
<td>For weekly scheduled workflow, the day of the week to run the workflow. For monthly scheduled workflows, the day of the month to run the workflow.</td>
</tr>
<tr>
<td>Repeat Interval</td>
<td>For periodical workflows, the interval between workflows, beginning from the Starting date and time.</td>
</tr>
</tbody>
</table>
Geneva ServiceNow ServiceNow Platform

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting</td>
<td>For periodical workflows, the first date and time to run the workflow. For scheduled workflows run once, the date and time to run the workflow.</td>
</tr>
</tbody>
</table>

Workflow activity updates (pinning)

Workflow administrators can *pin* a custom activity to prevent the system from automatically updating that activity when a new version is downloaded from the ServiceNow Store.

You can pin or unpin individual activities or set pinning within workflow properties that controls the versions used for all the activities in that workflow. This can result in two workflows using different versions of the same activity.

*Note:* Activity pinning and unpinning applies to the custom activities downloaded from the ServiceNow Store only, and does not apply to newly published activity definitions made locally on your instance. To make use of these locally updated custom activities, you must check out your workflow and manually add the activities.

Workflow activity level pinning

Set pinning at the activity level to control versioning for that activity only.

Role required: workflow_admin

The activity uses this setting unless it is overridden by a different setting in a workflow.

1. Navigate to Workflow > Workflow Editor.
2. Open the workflow containing the activity whose version you want to control.
3. Check out the workflow.
4. Right-click the activity and select one of these options from the context menu:
   - **Pin Activity:** Prevents the activity from automatically updating when a new version is available. Pinned activities are labeled *Out of date* when a new version is available.
   - **Unpin Activity:** Allows the activity to automatically update to the latest version. Unpinned activities are updated when a new version is available.
5. Publish the workflow.

Workflow level pinning

Pinning at the workflow level overrides individual settings for all activities in that workflow.

Role required: workflow_admin

This action does not change the setting on the activity itself, which might use a different setting in another workflow. The default setting in the workflow properties uses the pinning setting of each activity.

1. Navigate to Workflow > Workflow Editor.
2. Open a workflow and check it out.
3. Click the properties icon in the title bar.
4. In the Activities section, select one of the following options in the Activity pinning field:
   - **Set by activity:** Allows all activities in the workflow to use their own pinning settings. This is the default pinning option.
   - **Pin all activities:** Pins all activities in that workflow to their current version.
• **Unpin all activities**: Allows all activities in that workflow to be updated.

5. Publish the workflow.

Out of date workflow activities

Pinned activities become out of date when a new version of the activity is downloaded from the ServiceNow Store.

The workflow editor highlights a pinned activity with an orange indicator when it is out of date. Also, the **Out of date** option on the properties form is selected. To update a pinned activity, workflow designers must manually update the activity.

Figure 886: Activity out of date

Updated workflow activities

The workflow editor indicates when activities have been updated.

The workflow editor displays a green indicator when:

• An unpinned activity has been updated automatically
• A pinned activity has been updated manually
Manually update a workflow activity

You must manually update pinned activities.

Role required: workflow_admin

The workflow editor displays an orange indicator when the activity is out of date. To manually update an activity:

1. Navigate to Workflow > Workflow Editor.
2. Open the workflow containing an activity you want to manually update.
3. Check out the workflow.
4. Right-click a pinned activity and select Update Activity.
5. Publish the workflow.

Workflow movement with update sets

The system tracks workflows in update sets differently than other records because workflow information is stored across multiple tables.

Changes made to a workflow version are not added to the update set until the workflow is published, at which point the entire workflow is added into the update set. Update sets store workflows as a single Workflow [wf_workflow] record and only retain the latest version with the update type of Workflow.

Workflow update set migration use case - simple

Create a new workflow with no dependencies and then migrate the workflow in an update set.

1. User A selects Update Set A.
2. User A creates a new workflow called Workflow A.
3. User A publishes Workflow A.

A customer update set record is added to Update Set A containing an XML payload, including the published Workflow A and all activity dependencies. The XML payload also contains the workflow input variables associated with the workflow.
4. User A completes Update Set A and migrates it to the production instance.
5. Update Set A commits successfully.
6. Workflow A works as expected.

Workflow update set migration use case - subflow dependency (success)
Successfully edit and migrate an existing workflow and its dependent subflow.
1. User A selects Update Set B.
2. User A checks out Workflow A.
3. User A adds a subflow called Workflow B to Workflow A.
   Assume that Workflow B was previously published and migrated to the production instance.
4. User A publishes Workflow A.
   A customer update set record is added to Update Set B containing an XML payload, including the published Workflow A and all activity dependencies. The XML payload also contains the workflow input variables associated with the workflow.
5. User A completes Update Set B and migrates it to the production instance.
6. Update Set B commits successfully.
7. Workflow A works as expected with Workflow B as a subflow.

Workflow update set migration use case - subflow dependency (failure)
Edit and migrate an existing workflow from a test instance to a production instance that fails to run on the production instance because of a missing dependent subflow.
1. User A selects Update Set C.
2. User A checks out Workflow A.
3. User A adds a subflow called Workflow B to Workflow A.
   Assume that Workflow B was previously published, but has not been migrated to the production instance.
4. User A publishes Workflow A.
   A customer update set record is added to Update Set C containing an XML payload, including the published Workflow A and all activity dependencies. The XML payload also contains the workflow input variables associated with the workflow.
   Notably absent from Update Set C is the subflow called Workflow B. Workflow B was published before User A selected Update Set C.
5. User A completes Update Set C and migrates it to the production instance.
6. Update Set C commits with warnings.
7. Workflow A is invoked on the production instance with the following results:
   Workflow A fails the runtime validation check and is prevented from running on the production system. The system adds to the workflow context a workflow log entry detailing the cause of the failure, notably the absence of a dependent workflow.

To learn more about the validation checks on workflow dependencies and update sets see ValidateUpdateSetDependencies on page 3606.
Workflow update set migration use case - subflow dependency (risk)

Multiple users migrate a workflow from a test instance to a production instance without proper coordination. This use case can succeed, but only when each user understands the dependencies and properly migrates the dependent parts of the workflow to the new instance.

This example does not represent an update set failure, although update sets are most often blamed in this use case. Validation increases the visibility of workflow dependencies across multiple update sets and provides designers with better information. In most cases, the warnings do not prevent an action, but only identify risk. The designer is responsible for taking action on advice given in the validation checks.

1. User A selects Update Set C.
2. User A checks out Workflow A.
3. User A adds a subflow called Workflow B that returns a **User ID**.
   
   **Note:** Assume that Workflow B was previously published and migrated to the production instance.
4. User A uses the return value of Workflow B to generate approvals.
5. User B selects Update Set D.
6. User B checks out Workflow B (the subflow in Workflow A).
7. User B modifies the return value of the workflow by changing it from a **User ID** to a **String Message**.
8. User A publishes Workflow A.
   
   **Note:** A dialog box displays warnings associated with Workflow A and encourages User A to validate the workflow before publishing.
9. User A cancels publishing and validates Workflow A.
10. User A is warned that Workflow B was modified by a user in a different update set.
11. User A ignores this warning and publishes Workflow A.
    
    **Note:** A customer update set record is added to Update Set C containing an XML payload, including the published Workflow A and all activity dependencies. The XML payload also contains the workflow input variables associated with the workflow.
12. User A completes Update Set C and migrates it to the production instance.
13. Workflow A is invoked on the production instance and runs successfully using the older version of Workflow B already on the system.
14. User B publishes Workflow B.
   
   **Note:** User B is not warned of the Update Set C dependency, because the update set is no longer In progress. However, User B is informed via a dialog box that there are warnings associated with the workflow version and is instructed to validate Workflow B. If User B cancels publication and validates the workflow, User B is warned that there are workflows that use Workflow B as a subflow. Knowing the return value was changed, User B should test those workflows as well. See ValidateUpdateSetDependencies on page 3606 to understand the parameters of update set warnings.
15. User B finally publishes Workflow B.
    
    **Note:** A customer update set record is added to Update Set D containing an XML payload, including the published Workflow B and all activity dependencies.
16. User B completes Update Set D and migrates it to the production instance.

17. Update Set D commits without warnings.

18. Workflow A is invoked on the production instance and fails to run successfully, because the return value of Workflow B no longer generates a User ID.

Input variable movement

You can add input variables to existing workflows and add them to update sets.

When you submit the new variables, an entry is made into the current update set that reflects the addition of a variable to the Variables [var_dictionary] table. Unlike the workflow version that only writes to the update set when the workflow is published, the variables write individual update entries into the currently selected update set immediately upon submission.

Input variable movement use case - two input variables

An existing workflow already contains two input variables.

1. User A checks out the workflow.

2. User A adds two input variables.

   ServiceNow adds to the current update set one customer update record for each new variable.

   The current workflow now has 4 input variables: the two that were present prior to check out and the two new ones.

3. User A publishes the workflow.

   There are now three related customer update records: two for new variables, and one for the published workflow. The XML payload of the new workflow version now includes all input variable database entries. So while the two original input variables do not have individual customer update records, all four variables are migrated to the local instance with the payload of the newly published workflow version.

4. Verify variables included in a specific workflow on page 3655.

5. User A completes the update set.

6. Adding Input Variables - Success

   User A migrates and commits the update set to a local instance where the original workflow version had previously been committed.

   • The two existing input variables are already present because of the earlier version.
   • The system adds the two new input variables when the user commits the update set.
   • The system preserves the two legacy input variables on the instance receiving the update set. The update set does not overwrite these variables.
   • The new published workflow version uses all four variables.
   • The user tests the new workflow version and it runs as expected.

Verify variables included in a specific workflow

Follow the steps in this page to verify the variables that are included in a specific workflow.

1. Navigate to System Update Sets > Local Update Sets.

2. Select the active update set.

3. Select the customer update entry for the workflow.

4. View the XML Payload.

5. Search for the name of one of the columns or search for var_dictionary.
There is one var_dictionary entry for each input variable.

**Input variable removal**
Deleting workflow input variables, like insert and update actions, creates a customer update record in a user's current update set.

These deletions migrate to a new instance with the update set, regardless of whether the workflow that owns the input variables is published in the same update set. Plan carefully and use caution when editing a workflow and selecting update sets.

**Input variable removal risk**
An existing workflow already contains two input variables.

1. The workflow was migrated to a production instance with the two variables.
2. On a development instance, User A selects Update Set A and checks out the workflow.
3. User A removes one input variable and all references to it in the workflow.
   The system enters into Update Set A one customer update record reflecting the deletion of the input variable. No record is added for the new workflow version which no longer depends on the input. This does not happen until the workflow is published.
4. User A continues working on other features in Update Set A that need to be moved to production.
5. User A completes Update Set A and migrates it to the production instance without publishing the workflow.
   The update set entry that deletes the workflow input variable now applies to the production instance. The prior version of the workflow is running on this instance and still references the missing variable.

**Input variable removal solution**
When editing workflows, particularly when deleting input variables, be sure to use a single update set for all variable editing and workflow publishing.

If necessary, merge the update set into a more general set targeted for deployment after the workflow is published.

---

**Note:** If a workflow version is already running on a production system and input variables are deleted from a newer version, those deletions could affect transactions already running against the earlier version. Use extreme caution when deleting workflow input variables and plan the migration carefully.

**Input variable removal prevention**
Prior to publishing a workflow version, the system validates the workflow model to assist the designer in planning for deployment.

This validation warns of critical errors that can prevent a workflow from running successfully, but also warns of dependencies and conflicts in update sets. See ValidateUpdateSetDependencies on page 3606 for more details.

**Avoiding duplicate workflows**
Update sets manage the published state of all versions of a workflow prior to committing the workflow version on a local instance.

The last version of a workflow committed as an Insert or Update using an update set becomes the currently published version, regardless of the publishing sequence for the workflow versions.

**Commit a workflow in an update set**
Follow the steps in this page to commit a workflow in an update set.

1. Workflow A - Version 1 is created and published in Update Set A.
2. Update Set A is completed and migrated to a local instance.

3. When the update set is committed, the system sets all prior versions of Workflow A to published = false.
   
   In the first migration, there are no prior versions.

4. Workflow A - Version 1 becomes the only published version of the workflow.

Update set migration example

It is not possible to have multiple published versions as a result of update set commits. However, this does not eliminate risk, and care should be taken when migrating update sets.

Consider this example:

1. Workflow A - Version 1 is migrated and committed to the production instance.

2. Update Set B is created.

3. Update Set C is created.

4. Workflow A - Version 2 is published in Update Set B.

   A customer update record is added to Update Set B with the Version 2 payload.

   A customer update record is added to Update Set B with the Version 1 workflow left unpublished.

5. Update Set B is completed.

6. Workflow A - Version 3 is published in Update Set C.

   A customer update record is added to Update Set C with the Version 3 payload.

   A customer update record is added to Update Set C with the Version 2 workflow left unpublished.

7. Update Set C is completed.

8. Update Set C is migrated and committed to the production instance.

   Workflow A - Version 1 is set to unpublished.

   Workflow A - Version 2 update is skipped since Update Set B, which contains Version 2, was never migrated.

   Workflow A - Version 3 is committed and becomes the only published version of the workflow.

Update set migration risk

Update Set B is migrated and committed to the production instance.

1. Workflow A - Version 3 is set to unpublished.


3. Workflow A - Version 2 is committed and becomes the only published version of the workflow.

   The workflow has gone back a version, perhaps unintentionally. The regressed version becomes the currently published version.

Set the maximum number of workflow activities

Limit the number of executing activities contained in your workflow to prevent runaway processing loops.

The number of executing activities in a workflow context is the sum of the entries in the activity history list and the currently executing activities. When the workflow reaches the maximum number of executing activities allowed by this setting, the workflow stops. The default maximum setting is 100. Use a setting that is at least 10% greater than the total number of executing activities you anticipate in your workflow.
Make sure you include activities such as Turnstile and Rollback To that cause the workflow to repeat processing.

To change the maximum number of workflow activities:

1. Navigate to Workflow > Workflow Editor.
2. Check out a workflow.
3. In the title bar, click the Workflow Properties icon.
4. Click the Activities tab.
5. In the Max activity count field, enter a value that accommodates the expected number of executing activities for the workflow.
6. Click Update.

Workflow timelines

The system provides a timeline view of history activities associated with a workflow context.

Timelines display a linear calendar of activities, such as tasks and approvals, defined by their start and end dates. Each activity on the timeline is represented by a span, which is displayed as a horizontal, colored bar. Each span has a label and a tooltip that contains additional information about the activity. The left pane displays all the activities in the context (or contexts) in an expandable hierarchy. You can change the timeline’s perspective for a more granular view of the data.

**Note:** Workflow timelines reflect context history only and are not real-time gauges of workflow activity.

Use a timeline

By default, the timeline displays all activities and transitions requested when first opened.

Collapse any part of the hierarchy in the activity pane and the timeline adjusts automatically. Date/time and duration controls enable you to scale the timeline to view all the elements at once. To display a timeline, click a UI action within a Workflow Context record.

1. Navigate to Workflow > Live Workflows > Active Contexts.
2. Select a context.
3. In Related Links, click Show Timeline to display the timeline for the entire context.

   The timeline opens with all activities expanded and the view set to Max, which displays the entire timeline at the width of the pane. The title of the timeline is in the form Workflow context: <context name>, Requested Item: <requested item number>.
4. Use the Range Selectors at the top of the timeline to change the perspective.

The increments go from one day to one year. To limit the timeline to an increment between the start date of the first span and the end date of the last span, click **Max**.

5. Use the starting and ending calendar fields to select the timeline perspective.
These fields control the same perspective as the slider at the bottom of the timeline.

6. Use the pink slider at the bottom of the timeline to change the perspective.
1. Move the slider from right to left to view all the spans on a long timeline.

2. Adjust the end points of the slider to make arbitrary changes to the magnification.
   A narrow slider zooms in on the spans and provides a more detailed view of complex timelines. A wide slider pulls the view out and makes more of the timeline visible on the screen.

7. To focus the timeline view on selected activities, expand or collapse the activity tree.
   Spans not visible in the activity tree are not shown in the timeline pane.

8. Hover over an activity span to display a tooltip with information about the activity.
   This action highlights the activity in the activity pane.
9. Hover over the transition between two activities to highlight the activity and the predecessor activity in the activity pane.

10. Double-click a span to display a history record for that activity. History records show information such as the State and the starting and ending times.
Timeline for a selected activity

You can display a partial workflow timeline.

1. Navigate to **Workflow > Live Workflows > Active Contexts**.
2. Open a Workflow Context record.
3. On the **Workflow Activity History** related list, select one or more individual activities.
4. Click **Show Timeline** from the action menu.
   
   The resulting view is a snapshot of the timeline, showing only the selected activities and their transitions, if any.
5. Collapse the tree to confine the view even further.
6. To view a timeline displaying activities from different contexts:

   **Note:** You might use this feature to display a subflow's context with the parent workflow context.

   1. Navigate to **Workflow > Live Workflows > History**.
   2. Select individual history items from the list.
   3. Select the **Show Timeline** option from the actions menu.

      History items are arranged in a hierarchy in the activities pane under their contexts. The timeline title is **Multiple Contexts**. The timeline draws only the activities and relationships of the history items selected.

---

**View subflows in a workflow timeline**

In a workflow timeline, subflow spans appear as a different color than the activities of the main workflow.
Workflow error handling

The personalized JavaScript that users create in workflow activity variables is vulnerable to run-time syntax errors.

In the base system, workflow activities do not provide condition routing on the error state. As a result, the workflow progresses based on the state of the current record. For example, a workflow contains an Approval - User activity that uses an advanced script to add additional approvers. A syntax error in the script results in no approvers being added. Because a state of no approvers is a valid return, even without the syntax error, the approval activity is skipped and the workflow progresses along a positive path. However, this might not be a valid response for the workflow designer who does not want the workflow to progress along the positive path without approvers.

Workflow error handling detects and logs syntax errors and provides a state that the workflow designer can use to add error conditions to the workflow. Use error handling to locate syntax errors in advanced script fields for these workflow activities:

- Approval - User
- Approval - Group
- Catalog Task
- Create Task
- If
- Run Script
- Notification
Available error information

This information is available in a tooltip when you point to a workflow activity in an error state. This table also shows which activities support error exits.

Table 1249: Available error information

<table>
<thead>
<tr>
<th>Activity</th>
<th>Workflow log</th>
<th>Red error indicator</th>
<th>Activity state</th>
<th>Activity result</th>
<th>Fault description</th>
<th>Reroute on error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval - User</td>
<td>Yes</td>
<td>Yes</td>
<td>Error</td>
<td>Skipped</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Approval - Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog Task</td>
<td>Yes</td>
<td>No</td>
<td>Finished</td>
<td>none</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Create Task</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If</td>
<td>Yes</td>
<td>Yes</td>
<td>Error</td>
<td>none</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run Script</td>
<td>No</td>
<td>Yes</td>
<td>Error</td>
<td>error</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Notification</td>
<td>Yes</td>
<td>Yes</td>
<td>Error</td>
<td>error</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Tracking errors

Error handling provides visual cues within the workflow, error descriptions for activities in pop-ups, and detailed log records.

Error handling makes error detection and troubleshooting easier.

Table 1250: Error cues

<table>
<thead>
<tr>
<th>Cue</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banners</td>
<td>Look for an activity with a red banner, indicating that a syntax error has occurred in a script field. All activities that provide error handling, with the exception of Catalog Task and Create Task, display a red banner for this error.</td>
<td>Figure 889: Error handling banner</td>
</tr>
<tr>
<td>Cue</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Tooltips</td>
<td>Point to the activity displaying a red banner to view information about the error. A tooltip shows the State and Result of the activity and provides a brief Fault Description (except for task activities). Note that this approval continued as skipped despite the error given in the fault description. See Available error information on page 3665 for the information available to each activity.</td>
<td>Figure 890: Error handling tooltip</td>
</tr>
<tr>
<td>Cue</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Execution order  | Tooltip text in the Workflow Context graphical view displays the execution order of individual activities, which assists in troubleshooting. | To view the order in which a workflow activity was executed:  
1. Navigate to Workflow > Live Workflows > Active Contexts or All Contexts.  
2. Open the context you want to examine.  
3. Click Show Workflow.  
4. Hover the cursor over a finished or executing activity.  
   A tooltip appears showing error data, execution time, and the order in which that activity executed in the workflow. You can use this data to help troubleshoot activities in an error state. |
<table>
<thead>
<tr>
<th>Cue</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow log</td>
<td>View the log in the Workflow Context form for more information about the syntax error in the activity. Since task activities do not display a red banner when a syntax error has occurred, you must view the log if you suspect the workflow has not run properly. Examine the error description in the log, and then inspect the script in the activity named in the log.</td>
<td>To view the activity by name, navigate to Workflow &gt; Administration &gt; Properties and enable the Log workflow debug messages property. In this example, an activity named Manager approval (Approval - User) contains a script that is missing a semicolon.</td>
</tr>
</tbody>
</table>

Create an error condition exit

An administrator can reroute the workflow when a script error occurs by creating an error condition exit for specific activities within the workflow. This allows the workflow to process script errors in a predictable way and not create undesirable results.

1. Open and check out a workflow.
2. Right-click in the top portion of the activity for which you want to create an error exit.
3. Select **Add condition** from the context menu.
4. Add a condition exit with the following values:
   - **Name:** Error
   - **Condition:** activity.state=='faulted'
5. Click **Submit**.
   The **Error** exit appears on the activity.
6. Connect the **Error** exit to another activity for handling the error state, such as **Notification** or **Log Message**.

---

**Reconfigure an approval condition**

Approval activities react differently to script errors than the other activities.

Approval activity script errors can prevent an approval from being processed successfully. This, in turn, can cause the approval to complete as **Skipped**, which can appear to be an **Approved** state. To prevent this from happening, reconfigure the **Approved** exit as follows:

1. Open an approval workflow and make sure it is checked out.
2. On an approval activity, double-click the **Approved** exit.
3. Add the following script to the **Condition** field:
   ```javascript
   && activity.state != 'faulted'
   ```
   This prevents the activity from continuing down the normal path in an error state and ensures that **Skipped** or **Approved** is the desired state.
4. Click **Update**.
Workflow run time metrics

You can enable the collection of workflow run time metrics to determine whether workflows are performing as expected or consuming additional resources.

Outlying run times for a workflow are identified by comparing actual run times to an outlier range calculated with the outlier threshold and estimated run time defined in the workflow properties.

You can monitor the results of these metrics on the Workflow Operations Dashboard and custom homepages with workflow gauges.

Enable workflow run time metrics

Provide an estimated run time that can be compared to actual workflow run times.

For baseline workflows, you must also manually enable the collection of run time metrics. The system automatically enables the collection of run time metrics for new workflows.

To collect run time metrics for a workflow:

1. Navigate to Workflow > Workflow Editor.
2. Open and check out the workflow.
3. In the title bar, click the menu icon and select Properties.
4. In the Workflow Properties dialog box, click the Estimated Runtime tab.
5. To enable the collection of run time metrics, check that the Requires ERT option is selected.
6. Open a configuration from the Estimated Run Time column.
7. In Estimated Run Time, enter an initial estimate for the workflow's run time.
   The system compares this initial estimate to actual run time results to create outlier reports. The system can automatically update this field in certain circumstances. For more information, see Workflow estimated run time updates on page 3671. Workflow designers can also manually update this field.
8. In the Outlier Percentage Threshold for ERT field, enter the percentage deviation from the estimated run time that identifies an outlier workflow run time. For more information, see Workflow estimated run time updates on page 3671.
   The system uses a default value of 20.
9. Click Update.

Outlying workflow run times

Workflow run times are identified as outliers when they are longer or shorter than the outlier range that is computed for the workflow.

The outlier range is automatically computed with the Estimated Run Time and Outlier Percentage Threshold for ERT values in the workflow properties. These values are used in the following formulas.

Table 1251: Formulas for computing workflow outlier ranges

<table>
<thead>
<tr>
<th>Value computed</th>
<th>Computation used</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlier Value</td>
<td>Estimated Run Time * (Outlier Percentage Threshold for ERT / 100)</td>
<td>10 seconds * (20 / 100)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 seconds * 0.2 = 2 seconds</td>
</tr>
<tr>
<td>Outlier Range</td>
<td>(Estimated Run Time - Outlier Value) to (Estimated Run Time + Outlier Value)</td>
<td>(10 seconds - 2 seconds) to (10 seconds + 2 seconds) = 8 to 12 seconds</td>
</tr>
</tbody>
</table>
When a workflow runs within the outlier range, its estimated run time is automatically updated. If a workflow has an outlying run time, it appears in any outlier workflow gauges on the Workflow Operations Dashboard and custom home pages.

Workflow estimated run time updates

When a workflow runs within the outlier range, its estimated run time is automatically updated. The estimated run time is updated with the cumulative moving average of the latest run time value in relation to previous run times. The computed value is rounded to the nearest second and stored as a GlideDateTime.

For example:

<table>
<thead>
<tr>
<th>Data point</th>
<th>Latest value</th>
<th>Cumulative running average (CRA)</th>
<th>CRA after rounding to the nearest second</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 seconds</td>
<td>10 seconds</td>
<td>10 seconds</td>
</tr>
<tr>
<td>2</td>
<td>12 seconds</td>
<td>11 seconds</td>
<td>11 seconds</td>
</tr>
<tr>
<td>3</td>
<td>9 seconds</td>
<td>10.333 seconds</td>
<td>10 seconds</td>
</tr>
</tbody>
</table>

**Note:** Because the system rounds to the nearest second, the calculation is less precise with short durations.

You can also manually update the estimated run time in the workflow properties.

Troubleshoot workflows

Troubleshooting tools for workflows enable administrators to isolate execution paths, compare contexts, and track incomplete activities.

The workflow timeline provides a visual representation of the workflow, including transitions and the elapsed time for each activity. A troubleshooting tool for highlighting execution paths helps users perform forensics on a workflow. The highlighting feature can group multiple execution paths in various colors and can isolate rollback processing. Use the workflow highlighter to isolate incomplete tasks and approvals. You can also enable a workflow performance timing feature to troubleshoot slow workflows.

Execution path troubleshooting in timelines

Use the controls in a workflow timeline to isolate specific execution paths or compare multiple execution paths over time.

Timelines show how the activities in the workflow progressed in relation to one another over time. Isolate execution paths and follow transition lines between subflows and the main flow. Processing times provide a view of the workflow that you cannot get from the workflow diagram. Tooltips give precise information about each activity, such as duration.
Use the Workflow Operations Dashboard

On the Workflow Operations Dashboard, view and add gauges to help you monitor workflows. Review the performance of workflows and determine which workflows must be improved.

Role required: workflow_admin

The different levels of access are:

- **View**: View the dashboard and refresh gauges.
- **Customize**: Refresh, add, delete, and rearrange gauges.

On the Workflow Operations Dashboard, you can click gauge elements to view the records they represent. You can also add workflow gauges that are not displayed by default.

1. To view the Workflow Operations Dashboard, navigate to **Workflow > Operations > Workflow Operations Dashboard**.

   The default gauges on the dashboard include:
   - Workflow contexts providing run time metrics
   - Active workflows started during a specified time period
   - Workflows run between yesterday and today
   - Workflows by state for the current month
   - Aged workflow contexts for the past month
2.

To find and add more workflow gauges, click the **Add Content** icon ( ) in the upper left corner of the dashboard.

If a workflow consistently appears as an outlier and there is no error causing the run time values, you can use its historical run time values to calculate a new estimated run time value. Then update the estimated run time in the workflow properties.

**Workflow gauges**

Multiple gauges are available to help you review the performance of workflows. You can add these gauges to the Workflow Operations Dashboard or custom homepages.

<table>
<thead>
<tr>
<th>Table 1252: Workflow gauges</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERT Dashboard Controls &gt;</strong></td>
<td></td>
</tr>
<tr>
<td>Outlier Finished Workflows for ERT (Percentage Outlier)</td>
<td>Identifies workflows that finished, but did not finish within estimated runtime (ERT) values. Enter an outlier percentage to see workflows that ran outside the specified runtime range.</td>
</tr>
<tr>
<td>Outlier Long Running Workflows for ERT (Percentage Outlier)</td>
<td>Displays workflows running longer than the configured runtime threshold.</td>
</tr>
<tr>
<td><strong>Workflow Dashboard &gt;</strong></td>
<td></td>
</tr>
<tr>
<td>Workflows Without Current Record</td>
<td>Displays workflow contexts that do not have an associated current record.</td>
</tr>
<tr>
<td><strong>Gauges &gt; Workflow Context</strong></td>
<td></td>
</tr>
<tr>
<td>Number of Active Workflows Started Hourly Over Time (Yesterday)</td>
<td>Displays the total number of running workflows per hour over a given time period. By default, it displays the number of workflows that ran per hour over the previous day (yesterday).</td>
</tr>
<tr>
<td>Outlier Finished Workflows Not Cumulated to ERT</td>
<td>Displays workflow contexts that are finished outside the estimated runtime outlier value and not cumulated to estimated run time value.</td>
</tr>
<tr>
<td>Running Workflow Contexts</td>
<td>Displays the total number of running workflows.</td>
</tr>
<tr>
<td>Successfully Finished Workflows Cumulated to ERT</td>
<td>Displays the total number of successfully completed workflows whose running duration is cumulated to the estimated run time value.</td>
</tr>
<tr>
<td>Workflows by State (This Month)</td>
<td>Displays the total number of workflows run in a month grouped by the current state.</td>
</tr>
<tr>
<td>Aged Workflow Contexts (Running Since Last Month)</td>
<td>Displays the total number of workflow contexts running for a given period of time by workflow name. By default, it displays the total number of workflow contexts running over the last month.</td>
</tr>
<tr>
<td>Workflows Run Between Yesterday and Today (by Table)</td>
<td>Displays workflows that have run in the last day grouped by table name.</td>
</tr>
</tbody>
</table>
Execution history table

Highlight execution paths and rollbacks to locate activities that may have been left in an unresolved state. Rollbacks, cancels, and deletions during the execution of a workflow can prevent some activities from fully completing. Use highlighting in the execution history table to determine which activities in the workflow were left in an unresolved state.

1. Run the workflow.
2. Navigate to Workflow > All Contexts.
3. Select a context to troubleshoot.
4. In the Workflow Context form, select the Workflow Activity History related list.
5. Right-click an activity and select Workflow Debug > Toggle Execution Path Highlighting from the context menu.

All activities in that execution path are highlighted in a color selected by the platform. The debug path shows all activities that completed successfully during the workflow.

6. Right-click a rollback activity and select Workflow Debug > Toggle Rollback Highlighting from the context menu.

The platform highlights the rollback path (restarted activities) in a different color. Each color represents a group of activities that were part of the same rollback execution. The highlighting includes the activity that initiated the rollback. If you right-click an activity that was not part of a rollback, no rows are highlighted.

Note: The rollback activity itself appears in both execution path and rollback highlighting.
7. To remove highlighting, right-click in the list and select an option to clear execution path or rollback highlighting.
   You can clear individual rollback paths or all rollback highlighting.

Workflow performance timing

The workflow engine can generate detailed performance timing data that is useful for troubleshooting slow workflows.

An administrator must enable this functionality.

When workflow performance timing is enabled, the workflow engine tracks key performance data, including various execution speed metrics. The Workflow Timing [wf_workflow_timing] table stores the data, with a record for each workflow context. The workflow timing record is updated when the workflow engine completes the workflow, waits for an activity to complete, or otherwise exits the workflow execution.

Enable workflow performance timing

Workflow performance timing is disabled by default. You can create a system property to enable it.

Role required: admin

*Add a property using sys_properties.list* on page 1411 with the following specifications.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.workflow.show_timing</td>
</tr>
</tbody>
</table>
Workflow activities

A workflow activity defines an individual action that a workflow performs, such as adding a record or sending a notification. An activity can succeed or fail and the outcome can determine what the workflow does next.

Each activity has properties you can set to determine the details of how the activity behaves.

For information about configuring different types of activities, see *Workflow activities reference* on page 3676.

Workflow activities reference

Workflow activity reference, organized by category.

Each activity performs a different task, such as running a script, sending notifications, or requesting approvals.

Approval and rollback activities

Approval and rollback activities generate and manage approvals. Not all workflows can include approval activities. For more information, read *Approval and rollback workflow activities* on page 3680.

Table 1253: Approval and rollback activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval Action workflow activity</td>
<td>The Approval Action activity performs an approval action on the current task.</td>
</tr>
<tr>
<td>Approval Coordinator workflow activity</td>
<td>The Approval Coordinator activity manages the overall approval that results from one or more approval activities contained within it.</td>
</tr>
<tr>
<td>Approval - Group workflow activity</td>
<td>The Approval - Group activity creates approval records for each member of a specified group.</td>
</tr>
<tr>
<td>Approval - User workflow activity</td>
<td>The Approval - User activity creates one or more individual user approvals.</td>
</tr>
<tr>
<td>Generate workflow activity</td>
<td>The Generate activity immediately creates task or approval records from any task or approval activities placed after the Generate activity in the workflow path.</td>
</tr>
<tr>
<td>Manual Approvals workflow activity</td>
<td>The Manual Approvals activity watches and manages any approvals that users add manually outside of the workflow process.</td>
</tr>
<tr>
<td>Activity</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Rollback To workflow activity</strong> on page 3702</td>
<td>The <strong>Rollback To</strong> activity transitions directly to the activity specified by the outgoing transition line arrow.</td>
</tr>
</tbody>
</table>

### Condition activities

Condition activities provide conditional branching and logical operations for workflows.

**Table 1254: Condition activities**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If workflow activity</strong> on page 3703</td>
<td>The <strong>If</strong> activity checks a condition or script to determine if a <strong>Yes</strong> or <strong>No</strong> transition should be taken.</td>
</tr>
<tr>
<td><strong>Switch workflow activity</strong> on page 3704</td>
<td>The <strong>Switch</strong> activity checks if the value of a passed field or variable is equivalent to one of several case values.</td>
</tr>
<tr>
<td><strong>Wait for condition workflow activity</strong> on page 3708</td>
<td>The <strong>Wait for condition</strong> activity causes the workflow to wait at this activity until the current record matches the specified condition.</td>
</tr>
<tr>
<td><strong>Wait for WF Event workflow activity</strong> on page 3709</td>
<td>The <strong>Wait for WF Event</strong> activity causes the workflow to wait at this activity until the specified event is fired.</td>
</tr>
</tbody>
</table>

### Notify activities

Notify activities manage calls and SMS messages in Notify.

**Table 1255: Notify activities**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Forward Call workflow activity</strong> on page 2917</td>
<td>The <strong>Forward Call</strong> activity forwards a Notify call to an E.164-compliant phone number.</td>
</tr>
<tr>
<td><strong>Input workflow activity</strong> on page 2918</td>
<td>The <strong>Input</strong> activity creates a phone menu by presenting a list of options on a Notify call.</td>
</tr>
<tr>
<td><strong>Hangup workflow activity</strong> on page 2919</td>
<td>The <strong>Hangup</strong> activity disconnects an active Notify phone call.</td>
</tr>
<tr>
<td><strong>Play workflow activity</strong> on page 2920</td>
<td>The <strong>Play</strong> activity plays a sound file on a Notify call.</td>
</tr>
<tr>
<td><strong>Record workflow activity</strong> on page 2920</td>
<td>The <strong>Record</strong> workflow activity records audio from a user on a Notify call.</td>
</tr>
<tr>
<td><strong>Reject workflow activity</strong> on page 2921</td>
<td>The <strong>Reject</strong> workflow activity rejects an incoming Notify call.</td>
</tr>
<tr>
<td><strong>Say workflow activity</strong> on page 2921</td>
<td>The <strong>Say</strong> activity plays a message, using text-to-speech, on a Notify call.</td>
</tr>
<tr>
<td>Activity</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Forward to Notify Client workflow activity on page 2922</td>
<td>The <strong>Forward to Notify Client</strong> workflow activity connects a phone call to a Notify WebRTC client.</td>
</tr>
<tr>
<td>Call workflow activity on page 2915</td>
<td>The <strong>Call</strong> activity makes outbound phone calls using a Notify workflow.</td>
</tr>
<tr>
<td>Join Conference Call workflow activity on page 2914</td>
<td>The <strong>Join Conference Call</strong> connects an incoming or outgoing call to a Notify conference call.</td>
</tr>
<tr>
<td>Send SMS workflow activity on page 2916</td>
<td>The <strong>Send SMS</strong> activity sends short text messages to users’ phones using Notify.</td>
</tr>
<tr>
<td>Queue workflow activity on page 2923</td>
<td>The <strong>Queue</strong> activity places an active Notify call in a queue.</td>
</tr>
</tbody>
</table>

**Notification activities**

Notification workflow activities notify users of events that occur during the workflow.

Table 1256: Notification activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Event workflow activity on page 3719</td>
<td>The <strong>Create Event</strong> activity adds an event to the event queue, but does not immediately fire the event.</td>
</tr>
<tr>
<td>Notification workflow activity on page 3720</td>
<td>The <strong>Notification</strong> activity sends an email or SMS message to specified users or groups.</td>
</tr>
</tbody>
</table>

**Timer activities**

Timer activities pause workflows for set periods of time.

Table 1257: Timer activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLA Percentage Timer workflow activity on page 3721</td>
<td>The <strong>SLA Percentage Timer</strong> activity pauses the workflow for a duration equal to a percentage of an SLA.</td>
</tr>
<tr>
<td>Timer workflow activity on page 3722</td>
<td>The <strong>Timer</strong> activity pauses the workflow for a specified period of time.</td>
</tr>
</tbody>
</table>

**Task activities**

Task activities create and modify workflow tasks.
### Table 1258: Task activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Worknote workflow activity on page 3727</td>
<td>The Add Worknote activity adds text to the Worknotes field of the current incident record.</td>
</tr>
<tr>
<td>Attachment Note workflow activity on page 3727</td>
<td>The Attachment Note activity adds a comment to a journal field containing a link that points towards an attachment.</td>
</tr>
<tr>
<td>Catalog Task workflow activity on page 3728</td>
<td>The Catalog Task activity creates a service catalog task record.</td>
</tr>
<tr>
<td>Create Task workflow activity on page 3731</td>
<td>The Create Task activity generates a record on any of the tables that extend Task [task].</td>
</tr>
</tbody>
</table>

### Utility activities

Utility workflow activities provide controls over the path of the workflow and other useful tools.

### Table 1259: Utility activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch workflow activity on page 3734</td>
<td>The Branch activity splits the workflow into multiple transition paths from a single activity.</td>
</tr>
<tr>
<td>Join workflow activity on page 3735</td>
<td>The Join activity unites multiple execution paths into one transition.</td>
</tr>
<tr>
<td>Lock workflow activity on page 3735</td>
<td>The Lock activity prevents other instances of this workflow from continuing past this activity until the lock is released.</td>
</tr>
<tr>
<td>Log Message workflow activity on page 3738</td>
<td>The Log Message activity writes a message to the workflow log.</td>
</tr>
<tr>
<td>Log Trace Message workflow activity on page 3738</td>
<td>The Log Trace Message activity writes a trace message to the workflow log.</td>
</tr>
<tr>
<td>REST Message workflow activity on page 3738</td>
<td>The REST Message activity enables an administrator to override the REST endpoint or supply the variables configured in the REST Message module.</td>
</tr>
<tr>
<td>Return Value workflow activity on page 3739</td>
<td>The Return Value activity returns a value to a parent workflow, when run from a subflow.</td>
</tr>
<tr>
<td>Run Script workflow activity on page 3740</td>
<td>The Run Script activity runs a script you provide.</td>
</tr>
<tr>
<td>Set Values workflow activity on page 3740</td>
<td>The Set Values activity sets values on the current record.</td>
</tr>
<tr>
<td>SOAP Message workflow activity on page 3740</td>
<td>The SOAP Message activity uses SOAP messages defined in the System Web Services plugin and can call the messages using a MID Server.</td>
</tr>
</tbody>
</table>
### Turnstile workflow activity
The Turnstile activity limits how many times a workflow can pass through the same point.

### Unlock workflow activity
The Unlock activity releases a lock that was previously placed by the Lock activity.

### Subflow activities
Subflow activities run and manage workflows from a parent workflow.

**Table 1260: Subflow activities**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallel Flow Launcher workflow activity</strong></td>
<td>The Parallel Flow Launcher activity launches multiple subflows in parallel.</td>
</tr>
</tbody>
</table>

### Activities provided with Orchestration
The following activities are included with Orchestration.

- Active Directory activity pack
- Orchestration activities
- PowerShell activities
- Puppet activities

### Templates provided for creating custom activities
If Orchestration is active on your system, users with the proper roles can create custom activities using the ServiceNow. For information about the templates Orchestration provides for creating custom activities that you can upload to the ServiceNow Store, see

### Approval and rollback workflow activities
Approval and rollback activities generate and manage approvals.

Approval and rollback activities are not available in some workflows.

- With two exceptions, approval and rollback activities are only available when the workflow runs on a table that extends Task. The exceptions are the Approval - User and Approval Action activities, which are available globally.
- Approval and rollback activities are available only if approval engines are turned off for the table on which the workflow runs. If approval engines are enabled for the table, approval activities appear greyed out and cannot be selected. To learn more about how workflow and approval engines interact, read Approval workflow activities and approval engines on page 3762.

**Approval Action workflow activity**
The Approval Action activity performs an approval action on the current task.
Use this activity to mark the current task record as approved or rejected.

*Note:* When an **Approval Action** activity is used to mark a task approved, the activity marks all pending approvals as **No Longer Required**. This activity behaves differently from **Set Values** or **Run Script** when used to set the **Approval** field’s value.

**Results**

The result value of the activity is the final approval disposition selected by the approver. The result value can be **Approved** or **Rejected**. A workflow designer can assign a result value using the `activity.result` variable from within a script field of the activity.

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 1261: Approval Action activity input variables**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>The action to perform on the task. Options are:</td>
</tr>
<tr>
<td></td>
<td>• Mark task approved</td>
</tr>
<tr>
<td></td>
<td>• Mark task rejected</td>
</tr>
<tr>
<td></td>
<td>• Mark task requested</td>
</tr>
<tr>
<td></td>
<td>• Disregard pending approvals</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition runs after this activity.

**Table 1262: Approval Action activity conditions**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always</td>
<td>The event or condition that causes the approval to move to the next workflow step.</td>
</tr>
<tr>
<td>Error</td>
<td>The event or condition that generates an error.</td>
</tr>
<tr>
<td>Skipped</td>
<td>The event or condition that allows a skipped approval.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.
Table 1263: Approval Action activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine starts the execute function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Approval Coordinator workflow activity**

The Approval Coordinator activity manages the overall approval that results from one or more approval activities contained within it.

The Approval Coordinator contains one or more Approval - User, Approval - Group and Manual Approval activities. It waits for all approval activities contained within it before deciding whether it should complete with a result of approved or rejected.

When the Approval Coordinator activity completes, it sets all pending approvals created by any of the contained activities to No Longer Required. If a workflow calls a single user as an approver more than once, such as when a user is both a product- and an executive-approver, it skips any approvals for that user after the first.

**Results**

The result value of the Approval Coordinator activity depends on the approval actions taken by the approvers and the approval conditions specified in the Wait for field. Possible result values are:

- Approved
- Rejected
- Deleted
- Cancelled

**Input variables**

Input variables determine the initial behavior of the activity.
## Table 1264: Approval Coordinator activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait for</td>
<td>Indicate what to wait for to indicate that the approval activity is approved or rejected. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Any child activity to be approved:</strong> Any child activity (User, Group, or Manual Approval) that completes with a result of approved causes the Approval Coordinator activity to complete with a result of approved. Indicate what happens when any of the child activities completes with a result of rejected. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Reject the approval:</strong> Immediately complete the Approval Coordinator activity with a result of rejected.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Wait for other responses before deciding:</strong> Wait until we get other responses from other child activities before making an approval or rejection decision. This allows users to change their mind until a decision is made. In addition, if Wait for is set to <strong>Any child activity to approve</strong> then a single child activity completion with a result of approved will cause the Approval Coordinator activity to complete with a result of approved even if other child activities have completed with a result of rejected.</td>
</tr>
<tr>
<td></td>
<td>• <strong>All child activities to be approved:</strong> All child activities of the Approval Coordinator activity must complete with a result of approved to cause the Approval Coordinator activity to complete with a result of approved. Indicate what happens if any of the child activities completes with a result of rejected. (Options are described in previous bullet point.)</td>
</tr>
<tr>
<td></td>
<td>• <strong>The first approval or rejection from any child activity:</strong> The first child activity that completes with a result of approved or rejected causes the Approval Coordinator activity to complete with the same result.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Condition based on script:</strong> Call a script to determine how to manage an approval or rejection.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
Approval script | If the **Wait for** variable is set to **Condition based on script** this script is called to determine how to handle an approval or rejection. The script needs to set the variable **answer** to **approved** or rejected to indicate the overall approval status for this approval. When called, the following variable is available to the script:

```plaintext
counts.total = total number of child approval activities that are part of this approval
counts.approved = # of child approval activities that approved so far
counts.rejected = # of child approval activities that rejected so far
counts.requested = # of child approval activities that are pending approval
```
State | Description
---|---
Cancelled | This activity, or the workflow that contains this activity, was canceled.
Error | A JavaScript error occurred. Review the logs for error details.

**Approval - Group workflow activity**

The **Approval - Group** activity creates approval records for each member of a specified group.

The group approval is approved or rejected based on the user approvals, according to the logic specified in the **Wait For** field.

**Results**

The workflow designer can assign a result value using `activity.result` from within a script field of the activity. By default, the result value is the final approval disposition. This disposition depends on the approval actions taken by the approvers and the approval conditions specified in the **Wait for** or **When Anyone Rejects** fields. Possible result values are:

- Approved
- Rejected
- Deleted
- Cancelled

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 1267: Approval - Group activity input variables**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>Conditions which, if met, cause the group approval to be generated. If the conditions are not met, the approval is skipped.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Wait for</td>
<td>A choice between different approval logics to determine which individual approvals or rejections result in approval or rejection of the activity’s approval. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>An approval from each group:</strong> Any user from each group can approve and the first approval from each group causes the activity to complete with a result of approved (see below for how a rejection is handled).</td>
</tr>
<tr>
<td></td>
<td>• <strong>An approval from any group:</strong> Any user from any group can approve and the first approval from any group causes the activity to complete with a result of approved.</td>
</tr>
<tr>
<td></td>
<td>• <strong>An approval from everyone from all groups:</strong> All users from all groups must approve to cause the activity to complete with a result of approved (see below for how a rejection is handled).</td>
</tr>
<tr>
<td></td>
<td>• <strong>First response from each group:</strong> The first approval or rejection from any user in each group is used to indicate the state of the group approval (see below for how a rejection is handled).</td>
</tr>
</tbody>
</table>

Indicate what happens when any user rejects their approval request. Options are:

• **Reject the approval:** Immediately complete the activity with a result of rejected.

• **First response from any group:** The first approval or rejection from any user in any group causes the activity to complete with a result of approved or rejected.

• **Condition based on script:** Each time a user approves or rejects, the Approval script is called to determine if the activity should complete.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When anyone rejects</td>
<td>A choice between different approval logics to determine which individual rejections result in rejection of the activity's approval. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Reject the approval</strong>: Immediately complete the activity with a result of <strong>rejected</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Wait for other responses before deciding</strong>: Wait until we get other responses before making an approval or rejection decision. This allows users to change their mind until a decision is made.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: If Wait for is set to <strong>Anyone to approve</strong>, then a single approval causes the activity to complete with a result of <strong>approved</strong>, even if one or more users reject.</td>
</tr>
<tr>
<td>Groups</td>
<td>The groups whose approval will be requested. A particular group can be selected, or the tree icon can be used to select a group field from the current record. A user approval for each member of the group is automatically created when a Group approval record is created. If no group is selected, the activity automatically sets the approval to <strong>Approved</strong>.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to write a script for determining additional users to request approvals from. Use the <strong>Additional groups script</strong> to customize group approvals.</td>
</tr>
<tr>
<td>Due date based on</td>
<td>The due date fields are used to determine the values to use for setting <strong>Expected Start Time</strong> and <strong>Due Date</strong> for the task. The type determines how the due date is computed:</td>
</tr>
<tr>
<td></td>
<td>• <strong>A user specified duration</strong>: The duration is based on a user specified value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A relative duration</strong>: The duration is calculated from a relative duration (such as End of Next Business Day).</td>
</tr>
<tr>
<td></td>
<td>• <strong>A date/time or duration field</strong>: The duration is based on the value of a field on the current record.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Script</strong>: The duration is returned by a script.</td>
</tr>
<tr>
<td>Duration</td>
<td>The specific number of days and hours. This field is available only when <strong>Due date based on</strong> is <strong>A user specified duration</strong>.</td>
</tr>
<tr>
<td>Relative duration</td>
<td>The general number and length of business days. This field is available only when <strong>Due date based on</strong> is <strong>A relative duration</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Due date field</td>
<td>The date/time or duration field. This field appears when the <strong>Due date based on</strong> is <strong>A date/time or duration field</strong>.</td>
</tr>
<tr>
<td>Due date script</td>
<td>The script that sets 'answer' to the number of seconds for the duration. This field is available only when <strong>Due date based on</strong> is <strong>Script</strong>.</td>
</tr>
<tr>
<td>Schedule based on</td>
<td>The basic schedule the timer uses to count working hours. If a schedule is specified, the duration will only be considered for times that are specified on the schedule. For example, if the duration is 2 hours and the workflow begins at 4:00pm on a schedule that is 8am - 5pm, then it ends at 9:00am the next day. The options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>This workflow's schedule</strong>: The schedule uses workflow context date, time, and an optional <strong>Time zone based on</strong> value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A specific schedule</strong>: The schedule uses a pre-defined <strong>Schedule</strong> and an optional <strong>Time zone based on</strong> value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A schedule field</strong>: The schedule uses a value from a table and an optional <strong>Time zone based on</strong> value.</td>
</tr>
<tr>
<td>Schedule</td>
<td>The predefined <strong>Schedule</strong> from a list. This field available only when <strong>Schedule based on</strong> is <strong>A specific schedule</strong>.</td>
</tr>
<tr>
<td>Schedule field</td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when <strong>Schedule based on</strong> is <strong>A schedule field</strong>.</td>
</tr>
<tr>
<td>Time zone based on</td>
<td>The time zone for calculating the duration. The time zone may be based on:</td>
</tr>
<tr>
<td></td>
<td>• <strong>No time zone</strong>: Default. Workflow uses the GMT time zone.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A specific time zone</strong>: A predefined <strong>Time zone</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A time zone field</strong>: A <strong>Time zone field</strong> to track time duration from a field on the form.</td>
</tr>
<tr>
<td>Time zone</td>
<td>The predefined time zone. This field is available only when <strong>Time zone based on</strong> is <strong>A specific time zone</strong>.</td>
</tr>
<tr>
<td>Time zone field</td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when <strong>Time zone based on</strong> is <strong>A time zone field</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Additional groups script</td>
<td>If the <strong>Advanced</strong> check box is selected, this script is called to determine any additional group approvals to be created. The script needs to set the variable <code>answer</code> to a comma-separated list of group ids or an array of group ids to add as approver groups. For example:</td>
</tr>
<tr>
<td></td>
<td><code>answer = [];</code></td>
</tr>
<tr>
<td></td>
<td><code>answer.push('id1');</code></td>
</tr>
<tr>
<td></td>
<td><code>answer.push('id2');</code></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Approval script</td>
<td>If the <strong>Wait for</strong> variable is set to <strong>Condition based on script</strong> this script is called to determine how to handle an approval or rejection. The script needs to set the variable <strong>answer</strong> to approved or rejected to indicate the overall approval status for this approval.</td>
</tr>
<tr>
<td></td>
<td>This script is responsible for setting the approval state for each group that is part of this approval activity before returning the overall approval state for all of the groups.</td>
</tr>
<tr>
<td></td>
<td>When called, the following variables are available to the script for all the groups that are part of this approval activity:</td>
</tr>
</tbody>
</table>
|                  | counts.total = total number of groups that are part of this approval
|                  | counts.approved = # of groups that approved so far
|                  | counts.rejected = # of groups that rejected so far
|                  | counts.requested = # of groups that are pending approval
|                  | counts.not_requested = # of groups that are not pending approval
|                  | counts.not_required = # of groups that approval is not required
|                  | And for each group:
|                  | groups[group_id].total = total number of users that are part of this group's approval
|                  | groups[group_id].approved = # of users that approved so far
|                  | groups[group_id].rejected = # of users that rejected so far
|                  | groups[group_id].requested = # of users that are pending approval
|                  | groups[group_id].not_requested = # of users that are not pending approval
|                  | groups[group_id].not_required = # of users that approval is not required
|                  | groups[group_id].approvalIDs[state] = array of user ids that are at the specified approval state
|                  | Note: Iterate the groups using:
|                  | for (var id in groups) {                                                                                                                            |

© 2017 ServiceNow. All rights reserved. 3690
**Conditions**

The following conditions determine which transition runs after this activity.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved</td>
<td>The users from the groups have approved the request based on the <strong>Wait for</strong> rules.</td>
</tr>
<tr>
<td>Rejected</td>
<td>The users from the groups have rejected the request based on the <strong>Wait for</strong> rules.</td>
</tr>
<tr>
<td>Error</td>
<td>The event or condition that generates an error.</td>
</tr>
<tr>
<td>Skipped</td>
<td>The event or condition that allows a skipped approval.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine starts the <code>execute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Approval - User workflow activity**

The **Approval - User** activity creates one or more individual user approvals.

**Results**

The result value is the final approval disposition. This disposition depends on the approval actions take by the approvers and the approval conditions specified in the **Wait for** or **When Anyone Rejects** fields. Possible result values are:

- Approved
- Rejected
- Deleted
Input variables

Table 1270: Approval - User activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>Conditions which, if met, cause the individual approval to be generated. If the conditions are not met, the approval is skipped.</td>
</tr>
<tr>
<td>Wait for</td>
<td>A choice between different approval logics to determine which individual approvals result in approval of the activity's approval. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Anyone to approve</strong>: Any user can approve and the first approval causes the activity to complete with a result of approved.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Everyone to approve</strong>: All users must approve (see below for how a rejection is handled).</td>
</tr>
<tr>
<td></td>
<td>• <strong>First response from anyone</strong>: The first approval or rejection from any user causes the activity to complete.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Condition based on script</strong>: Each time a user approves or rejects, the Approval script is called to determine if the activity should complete.</td>
</tr>
<tr>
<td>When anyone rejects</td>
<td>A choice between different approval logics to determine which individual rejections result in rejection of the activity's approval. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Reject the approval</strong>: Immediately complete the activity with a result of rejected.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Wait for other responses before deciding</strong>: Wait until we get other responses before making an approval or rejection decision. This allows users to change their mind until a decision is made.</td>
</tr>
</tbody>
</table>

**Note**: Note that if **Wait for** is set to **Anyone to approve** then a single approval will cause the activity to complete with a result of approved even if one or more users reject.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval Column</td>
<td>A string field for the name of the approval field on the table the workflow is running on. The default value is approval, which is the field on the Task table.</td>
</tr>
<tr>
<td></td>
<td>Note: Use the field's name, not its label.</td>
</tr>
<tr>
<td></td>
<td>Note: If using any custom approval column fields and approval column journals, use Set Value activities in the workflow to set the custom Approval column fields.</td>
</tr>
<tr>
<td>Approval Journal Column</td>
<td></td>
</tr>
<tr>
<td>Users</td>
<td>The users for the approval. Use the tree icon to select user reference fields from the current record to create approvals, such as ${assigned_to}. If no user is selected, the activity automatically sets the approval to Approved.</td>
</tr>
<tr>
<td></td>
<td>Note: Workflow only manages approval records generated by the Approval User activity. After starting the workflow, newly added approvals do not affect the workflow context.</td>
</tr>
<tr>
<td>Groups</td>
<td>Groups whose members should also receive approvals. Note that this is different than the Approval - Group activity, which creates a group approval in addition to the individual approvals. The tree icon can be used to select group reference fields from the current record to create approvals, such as ${assignment_group}.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to write a script for determining additional users to request approvals from.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Due date based on             | The due date fields are used to determine the values to use for setting Expected Start Time and Due Date for the task. The type determines how the due date is computed:  
  • A user specified duration: The duration is based on a user specified value.  
  • A relative duration: The duration is calculated from a relative duration (such as End of Next Business Day).  
  • A date/time or duration field: The duration is based on the value of a field on the current record.  
  • Script: The duration is returned by a script. |
| Duration                      | The specific number of days and hours. This field is available only when Due date based on is A user specified duration.                                                                                                                                                                |
| Relative duration             | The general number and length of business days. This field is available only when Due date based on is A relative duration.                                                                                                                                                               |
| Due date field                | The date/time or duration field. This field appears when the Due date based on is A date/time or duration field.                                                                                                                                                                      |
| Due date script               | The script that sets 'answer' to the number of seconds for the duration. This field is available only when Due date based on is Script.                                                                                                                                              |
| Schedule based on             | The basic schedule the timer uses to count working hours. If a schedule is specified, the duration will only be considered for times that are specified on the schedule. For example, if the duration is 2 hours and the workflow begins at 4:00pm on a schedule that is 8am - 5pm, then it ends at 9:00am the next day. The options are:  
  • This workflow's schedule: The schedule uses workflow context date, time, and an optional Time zone based on value.  
  • A specific schedule: The schedule uses a pre-defined Schedule and an optional Time zone based on value.  
  • A schedule field: The schedule uses a value from a table and an optional Time zone based on value. |
<p>| Schedule                      | The predefined Schedule from a list. This field available only when Schedule based on is A specific schedule.                                                                                                                                                                       |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schedule field</strong></td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when <strong>Schedule based on</strong> is <strong>A schedule field</strong>.</td>
</tr>
<tr>
<td><strong>Time zone based on</strong></td>
<td>The time zone for calculating the duration. The time zone may be based on:</td>
</tr>
<tr>
<td></td>
<td>• <strong>No time zone</strong>: Default. Workflow uses the GMT time zone.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A specific time zone</strong>: A predefined Time zone.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A time zone field</strong>: A Time zone field to track time duration from a field on the form.</td>
</tr>
<tr>
<td><strong>Time zone</strong></td>
<td>The predefined time zone. This field is available only when <strong>Time zone based on</strong> is <strong>A specific time zone</strong>.</td>
</tr>
<tr>
<td><strong>Time zone field</strong></td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when <strong>Time zone based on</strong> is <strong>A time zone field</strong>.</td>
</tr>
<tr>
<td><strong>Additional approvers script</strong></td>
<td>If the <strong>Advanced</strong> check box is selected, this script is called to determine any additional user approvals to be created. The script needs to set the variable <strong>answer</strong> to a comma-separated list of user ids and group ids or an array of user and group ids to add as approvers. For example:</td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>answer = [];</td>
</tr>
<tr>
<td></td>
<td>answer.push('id1');</td>
</tr>
<tr>
<td></td>
<td>answer.push('id2');</td>
</tr>
</tbody>
</table>
### Approval Script

If the **Wait for** variable is set to **Condition based on script** this script is called to determine how to handle an approval or rejection. The script needs to set the variable **answer** to approved or rejected to indicate the approval status for this approval. When called, the following information is available:

```plaintext
counts.total = total number of users that are part of this approval
counts.approved = # of users that approved so far
counts.rejected = # of users that rejected so far
counts.requested = # of users that are pending approval
counts.not_requested = # of users that are not pending approval
counts.not_required = # of users that approval is not required
```

### Conditions

The following conditions determine which transition runs after this activity.

**Table 1271: Approval - User activity conditions**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved</td>
<td>The users approved the request based on the <strong>Wait for</strong> rules.</td>
</tr>
<tr>
<td>Rejected</td>
<td>The users rejected the request based on the <strong>Wait for</strong> rules.</td>
</tr>
<tr>
<td>Error</td>
<td>The event or condition that generates an error.</td>
</tr>
<tr>
<td>Skipped</td>
<td>The event or condition that allows a skipped approval.</td>
</tr>
</tbody>
</table>

### States

The activity state tells the workflow engine what to do with the activity.
Table 1272: Approval - User activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine starts the <code>execute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

Generate workflow activity

The Generate activity immediately creates task or approval records from any task or approval activities placed after the Generate activity in the workflow path.

By default, the workflow does not create any tasks or approvals until it reaches them in the workflow.

The Generate activity follows all transitions through the workflow to each activity. For each activity:

- If it is a task activity, creates the task and sets:
  - The State to Pending
  - The Expected Start Date
  - The Due Date

- If it is an approval activity, creates the approvals and sets:
  - The approval State to Not Requested
  - The Expected Start Date
  - The Due Date

Expected start dates and due dates are calculated based on the Expected Duration of all of the tasks and approvals between the Generate activity and the activity being updated. In the case of a branched path (between a Branch and Join activity), the longer duration will be used for any post-branch activities.

The Generate activity can be used more than once, and any tasks or approvals will be refreshed with updated information. This is useful in situations where the list of approvers or other important information is still editable while the workflow is in process and it may be necessary to update or correct the generated approvals or tasks.

To exclude a set of activities from the Generate activity, select the Skip during generate check box on any condition and its transitions will not be followed during the generate process. By default, the following conditions have the Skip during generate check box selected:

- Rejected (for any of the approval activities)
- No condition of If activity
- Continue condition of Turnstile activity
- Incomplete condition of Join activity
Input variables

Input variables determine the initial behavior of the activity.

Table 1273: Generate activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate approvals</td>
<td>If selected, approvals are created when running the <strong>Generate</strong> activity. If cleared, the approvals are used to compute their estimated duration, but no approvals are created.</td>
</tr>
<tr>
<td>Generate tasks</td>
<td>If selected, tasks are created when running the <strong>Generate</strong> activity. If cleared, the tasks are used to compute their estimated duration, but no tasks are created.</td>
</tr>
</tbody>
</table>

States

The activity state tells the workflow engine what to do with the activity.

Table 1274: Generate activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The activity is executing.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
</tbody>
</table>
Example

Figure 894: Generate workflow

In this example, the following approvals are generated:

- Approval-2
- Approval-3

Approval-4 is skipped since the Rejected condition of Approval-3 has Skip during generate selected.

Here is an example of using the Generate activity that describes the expected start and due dates:
In this example, if the **Generate** activity is run on Jan 1, 2016, the following expected start dates and due dates would be set for the generated tasks.

<table>
<thead>
<tr>
<th>Task</th>
<th>Expected Start Date</th>
<th>Reason</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1 (1 day)</td>
<td>Jan. 1, 2016</td>
<td>Jan. 2, 2016</td>
<td>Task 1 is 1 day</td>
</tr>
<tr>
<td>Task 2 (1 day)</td>
<td>Jan. 2, 2016</td>
<td>Task 3 is 2 days</td>
<td>Task 4 ends the latest before the Join</td>
</tr>
<tr>
<td>Task 3 (2 days)</td>
<td>Jan. 1, 2016</td>
<td>Jan. 3, 2016</td>
<td>Task 4 is 2 days</td>
</tr>
<tr>
<td>Task 5 (1 day)</td>
<td>Jan. 4, 2016</td>
<td>Jan. 5, 2016</td>
<td>Task 5 ends the latest before the Join</td>
</tr>
</tbody>
</table>

Notice that Task 5 starts on Jan. 4, 2016 since the longest path (based on due dates) to the **Join** is Task 3/Task 4.

**Manual Approvals workflow activity**

The **Manual Approvals** activity watches and manages any approvals that users add manually outside of the workflow process.

If there are no pending manual approvals when this activity executes, the activity immediately completes with a result of **approved**. This activity does not create approval records. Use this activity to pause the workflow when a user adds a manual approval to a record with an associated workflow. The workflow waits for the approval to be closed before proceeding.

**Results**

The workflow designer can assign a result value using `activity.result` from within a script field of the activity. By default, the result value of the activity is the final approval disposition determined by the approval actions take by the approvers. Possible result values are:

- Approved
- Rejected
- Deleted
- Cancelled
• Error

Input variables

Input variables determine the initial behavior of the activity.

Table 1275: Manual Approvals activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait for</td>
<td>Indicate what to wait for to indicate that the approval activity is approved or rejected. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Any manual user or group approval</strong>: Any user can approve and the first approval causes the activity to complete with a result of approved.</td>
</tr>
<tr>
<td></td>
<td>• <strong>All manual user or group approvals</strong>: All users must approve (see below for how a rejection is handled).</td>
</tr>
<tr>
<td></td>
<td>• <strong>The first response from any manual approval</strong>: The first approval or rejection from any user causes the activity to complete.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When anyone rejects</td>
<td>Indicate what happens when any user rejects their approval request. Options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Reject the approval</strong>: Immediately complete the activity with a result of rejected.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Wait for other responses before deciding</strong>: Wait until we get other responses before making an approval or rejection decision. This allows users to change their mind until a decision is made.</td>
</tr>
<tr>
<td></td>
<td>In addition, if <strong>Wait for</strong> is set to <strong>Anyone to approve</strong> then a single approval will cause the activity to complete with a result of approved even if one or more users reject.</td>
</tr>
</tbody>
</table>

Conditions

The conditions determine which transition runs after this activity.

Table 1276: Manual Approvals activity conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved</td>
<td>The users from the groups have approved the request based on the <strong>Wait for</strong> rules.</td>
</tr>
<tr>
<td>Condition</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Rejected</td>
<td>The users from the groups have rejected the request based on the <strong>Wait for</strong> rules.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.

**Table 1277: Manual Approvals activity states**

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine starts the <code>execute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Rollback To workflow activity**

The **Rollback To** activity transitions directly to the activity specified by the outgoing transition line arrow.

**Note:** The **Rollback** activity is deprecated starting with the Eureka release. Use the **Rollback To** activity instead.

**Rollback To** determines which activities to reset based on the actual workflow sequence (transition line attachments) of activities between itself and the transitioned to activity, not the execution order. **Rollback To** then marks all the approvals that have transitioned between the rollback and the transitioned to activity as **Not Yet Requested** and the tasks as either **Open** or **Pending**.

Use the **Rollback To** activity for all workflows in which multiple rollbacks are required. **Rollback To** has no variables.

**Conditions**

The conditions determine which transition runs after this activity.

**Table 1278: Rollback To activity conditions**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always</td>
<td>The event or condition that causes the approval to revert to the previous workflow step.</td>
</tr>
<tr>
<td>Error</td>
<td>The event or condition that generates an error.</td>
</tr>
<tr>
<td>Condition</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Skipped</td>
<td>The event or condition that allows a skipped approval.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.

**Table 1279: Rollback To activity states**

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine starts the <code>execute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Condition workflow activities**

Condition activities provide conditional branching and logical operations for workflows.

*If workflow activity*

The *If* activity checks a condition or script to determine if a *Yes* or *No* transition should be taken.

If the workflow creator specifies both the *Condition* and the *Advanced* script, both must evaluate successfully for activity to take the *Yes* transition.

**Results**

The workflow designer can assign a result value using `activity.result` from within the *Script* field on the activity record. By default, the result value of the activity is the final result of the condition or script specified. Possible result values are:

- Yes
- No

**Input variables**

The following variables determine the behavior of the activity.
Table 1280: If activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>If specified and the current record matches the condition, the <strong>Yes</strong> transition is taken.</td>
</tr>
<tr>
<td>Advanced and Script</td>
<td>To specify a script, select the <strong>Advanced</strong> check box. You may then enter a script that is evaluated. If your script sets the variable answer to yes, then the <strong>Yes</strong> transition is taken. Otherwise, the <strong>No</strong> transition is taken.</td>
</tr>
</tbody>
</table>

**Conditions**

The following conditions determine which transition comes after the activity.

Table 1281: If activity conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Taken when the condition, if specified, matches and the <strong>Advanced</strong> script, if specified, returns yes.</td>
</tr>
<tr>
<td>No</td>
<td>Taken when either the condition does not match or the <strong>Advanced</strong> script, if specified, returns no.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.

Table 1282: If activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the <code>onExecute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Switch workflow activity**

The **Switch** activity checks if the value of a passed field or variable is equivalent to one of several case values.
The switch activity behavior is similar to a switch statement in Java.

When creating a switch activity, select a **Variable** or **Field** to check against activity conditions for a matching field value. When the value passed in this variable or field matches the value defined in the **Condition** field of the activity condition, the workflow progresses through that activity condition.

**Results**

The variable or field selected in the **Variable** or **Field** activity variable determines the possible result values.

**Input variables**

The following variables determine the behavior of the activity.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Select <strong>Variable</strong> or <strong>Field</strong> as the type of value to check against available conditions. This selection sets the label and available options for the other field.</td>
</tr>
</tbody>
</table>
| **Variable or Field**| Select the source of the value compared against the switch activity conditions. The field label and available options depend on the **Type** selection.  
  • **Variable**: select any service catalog variable.  
  • **Field**: select any field from the **Table** defined in the workflow properties. |

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the <code>onExecute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>State</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Example**

You can create a switch activity that sets different field values on an incident based on the Assignment group of the incident record.

![Figure 896: Switch activity example](image)

The **Field** selected is the incident **Assigned to** field.
If the value of the Assigned to field of the workflow-triggering incident is Service Desk or Field Services, the workflow populates values on the incident record before continuing. If the Assigned to value is Hardware, the workflow continues without populating any field values.

Condition type (Standard, Else, or Error) is used. For more information, see Manage workflow activity conditions on page 3759 and Activity result value on page 3761.
The **Wait for condition** workflow activity causes the workflow to wait at this activity until the current record matches the specified condition.

The workflow evaluates the **Wait for condition** activity each time the current record is updated. Use this activity to pause a workflow indefinitely until a particular criteria is met by a record update. To pause a workflow for a timed duration see *Timer workflow activities* on page 3721.

For workflow to consider the condition met, all conditions specified – whether in the builder or in a script – must be true.

### Note:

---

**Figure 898: Hardware group condition**

**Wait for condition workflow activity**

The **Wait for condition** activity causes the workflow to wait at this activity until the current record matches the specified condition.

The workflow evaluates the **Wait for condition** activity each time the current record is updated. Use this activity to pause a workflow indefinitely until a particular criteria is met by a record update. To pause a workflow for a timed duration see *Timer workflow activities* on page 3721.

For workflow to consider the condition met, all conditions specified – whether in the builder or in a script – must be true.
A Wait for condition activity should only be used to wait for an external event such as a record update, and not one from a workflow setting a value. If you have a workflow setting a value and want to wait for that same field to be seen as ‘changed,’ try inserting a one-second timer.

**Results**

The workflow designer can assign a result value using `activity.result` from within a script field of the activity. The activity transitions when the result value is true.

**Input variables**

The following variables determine the behavior of the activity.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>The workflow is paused at this activity until this condition matches the current record.</td>
</tr>
<tr>
<td>Condition script</td>
<td>If specified, the workflow is paused at this activity until this script sets the <code>answer</code> variable to true.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the <code>onExecute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Wait for WF Event workflow activity**

The **Wait for WF Event** activity causes the workflow to wait at this activity until the specified event is fired.

Use this activity to wait for another activity to fire an event. Events from other activities are fired in a script using the `workflow.fireEvent('eventName')` API call.
Results
The workflow designer can assign a result value using `activity.result` from within a script field of the activity. This activity transitions when the specified event fires.

Input variables
The following variables determine the behavior of the activity.

Table 1287: Wait For WF Event activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait for Event</td>
<td>An event name to trigger the workflow.</td>
</tr>
</tbody>
</table>

States
The activity state tells the workflow engine what to do with the activity.

Table 1288: Wait For WF Event activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the <code>onExecute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

Notify workflow activities
Notify activities manage calls and SMS messages in Notify.

All Notify workflows that manage a phone call must run on the `notify_call` table. Call-related Notify activities can be added only to workflows where the `Table` field value is `Notify Call [notify_call]`.

You can identify if an outbound call was answered by a human or by an answering machine from within a Notify workflow by evaluating the `current.is_human` variable, such as with an `If` workflow activity. This variable is set by the telephony provider when an outbound call is answered. This variable is always true for inbound calls.

Note: Do not add a `Timer` activity between multiple Notify activities that interact with active phone calls.
**Important:** When creating a Notify workflow, set the workflow **If condition matches** field to **None**. Notify controls which workflow to run based on the configured number groups.

*Forward Call workflow activity*

The **Forward Call** activity forwards a Notify call to an E.164-compliant phone number.

If the person receiving a forwarded call hangs up, the **forward call** activity completes and transitions to the next activity. Any further Notify activities in the workflow run for the caller only.

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 1289: Input variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone number to call</td>
<td>Enter the phone number to forward the call to.</td>
</tr>
<tr>
<td>Timeout (in seconds)</td>
<td>Enter the amount of time to wait for the call to be answered before hanging up.</td>
</tr>
<tr>
<td>Record</td>
<td>Select this check box to record the conversation.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. The **forward call** activity does not specify any conditions by default.

You can add an error condition to this activity. The activity transitions through the error condition if the phone number to call is invalid.

*Input workflow activity*

The **Input** activity creates a phone menu by presenting a list of options on a Notify call.

**Input Variables**

Input variables determine the initial behavior of the activity.

**Table 1290: Input Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of digits</td>
<td>Specify the maximum number of digits the caller can enter. A caller can enter fewer digits than the maximum and press the <strong>Finish key</strong> to complete the entry.</td>
</tr>
<tr>
<td>Finish key</td>
<td>Specify the key a caller can press on their phone when finished selecting a menu option.</td>
</tr>
<tr>
<td>Timeout (in seconds)</td>
<td>Specify the amount of time to wait before closing the menu automatically when the caller does not select a menu option.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to build the phone menu, instead of using the activity conditions.</td>
</tr>
<tr>
<td>Script</td>
<td>Define the script to build the phone menu. The script must specify an answer variable as a JavaScript object with the following format:</td>
</tr>
</tbody>
</table>

```javascript
answer = {
  1: {
    "say": "https://some_url.com/options/one.mp3",
    "myCustomData": "some data here"
  },
  2: {
    "say": "type 2 to speak to a representative",
    "language": "en-US",
    "myCustomData": "some more data here"
  }
};
```

The script may specify either a text-to-speech string and language code or the URL of a prerecorded message for each entry. You can also add optional attributes to store related information, such as myCustomData in the example above.

**Conditions**

The conditions determine the transition that comes after this activity.

The **input** activity does not specify any conditions by default. You must define conditions to build the phone menu. Each condition is one option on the phone menu. Notify reads the text from each condition **Name** to the caller, up to 100 characters per condition.

You can specify a language for each condition by prefixing the message with the language code, in the format `xx-XX:<Message>`. For example, add `fr-CA:` for Canadian French. Available languages are stored on the Notify Language [notify_language] table.

The condition that the activity transitions through depends on the digits entered by the caller. Set the condition **Condition value** to `parseInt(workflow.scratchpad.digits) == <expected digits>`. For example, to transition through a condition when the caller presses the number 3, set the **Condition** to `parseInt(workflow.scratchpad.digits) == 3`.

You can add an error condition to this activity. The activity transitions through the error condition if the advanced script returns an invalid value, or if the text to say for a condition is empty.

**Scratchpad Entries**

The activity uses the workflow scratchpad to write persistent values.
Table 1291: Values written to scratchpad

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow.scratchpad.digits</td>
<td>The digits entered by the caller, as a string.</td>
</tr>
<tr>
<td>workflow.scratchpad.menu&lt;activity name&gt;</td>
<td>The entire answer variable, if using the advanced script option. You can access this menu from other activities after this activity successfully executes. For example, if the activity name is choices, you can access values from the menu using `var previousActivity = &quot;choices&quot;; var choicesMenu = workflow.scratchpad.menu[previousActivity]; var menuItem = choicesMenu[workflow.scratchpad.digits]; // Selects the menu item based on the caller’s input. var selectedValue = menuItem.myCustomData; //get the custom data for the selected menu item.</td>
</tr>
</tbody>
</table>

Hangup workflow activity

The **Hangup** activity disconnects an active Notify phone call.

You can use the **hangup** activity to disconnect only calls that have been answered. Use the **reject** activity to disconnect calls that have not been answered.

Play workflow activity

The **Play** activity plays a sound file on a Notify call.

**Input Variables**

Input variables determine the initial behavior of the activity.

Table 1292: Input Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Enter the URL of a sound file to play. If the URL is inaccessible, or if the audio file mime type is not supported by the telephony provider, the <strong>play</strong> activity is skipped.</td>
</tr>
<tr>
<td>Loop</td>
<td>Enter the number of times the sound file should play.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. The **play** activity does not specify any conditions by default.
You can create an error condition to handle errors with the sound file. The activity transitions through the error condition if the specified sound file is not available, or has an unsupported mime type. Supported mime types are listed on the Notify Audio MIME Types [notify_mime_type] table.

Record workflow activity
The Record workflow activity records audio from a user on a Notify call.

Input Variables

Input variables determine the initial behavior of the activity.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max duration (in seconds)</td>
<td>Enter the maximum duration, in seconds, allowed for a recording.</td>
</tr>
<tr>
<td>Timeout (in seconds)</td>
<td>Enter the amount of time to wait before ending a recording automatically when the caller is silent.</td>
</tr>
<tr>
<td>Finish Key</td>
<td>Specify the key a caller can press on their phone to end the recording.</td>
</tr>
</tbody>
</table>

Scratchpad Entries

The activity uses the workflow scratchpad to store persistent values.

The record activity adds the recording variable to the workflow scratchpad. This variable stores metadata about the recording, such as URL, ID, and duration. You can access the following values from this variable.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recording.recordingDuration</td>
<td>The duration of the recording, in seconds.</td>
</tr>
<tr>
<td>recording.from_number</td>
<td>The caller's phone number, including country code.</td>
</tr>
<tr>
<td>recording.notify_number</td>
<td>The Notify phone number used to respond to the call, including country code.</td>
</tr>
<tr>
<td>recording.recordingID</td>
<td>The ID used by the telephony provider to identify the recording.</td>
</tr>
<tr>
<td>recording.recordingURL</td>
<td>The URL from the telephony provider to access the recording.</td>
</tr>
</tbody>
</table>

Reject workflow activity

The Reject workflow activity rejects an incoming Notify call.

You can use the reject activity to disconnect only calls that have not yet been answered. Use the hang up activity to disconnect calls that have been answered.
Input variables

Input variables determine the initial behavior of the activity.

Table 1295: Input variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason</td>
<td>Select a reason for rejecting the call, such as busy.</td>
</tr>
</tbody>
</table>

Say workflow activity

The Say activity plays a message, using text-to-speech, on a Notify call.

Input variables

Input variables determine the initial behavior of the activity.

Table 1296: Input variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text to say</td>
<td>Specify the text to read.</td>
</tr>
<tr>
<td>Language</td>
<td>Select the language and locale to use when reading text.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to define the language and message, instead of using the Text to say and Language values.</td>
</tr>
<tr>
<td>Script</td>
<td>Define a script to set what text is read on the call. For example, to play an English-language message, return {language: 'en-US', text: 'Text to read'}.</td>
</tr>
</tbody>
</table>

Forward to Notify Client workflow activity

The Forward to Notify Client workflow activity connects a phone call to a Notify WebRTC client.

Input variables

Input variables determine the initial behavior of the activity.

Table 1297: Input variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Select the user to connect the call to.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to determine which client to connect to, instead of using the User variable.</td>
</tr>
</tbody>
</table>
### Conditions

The conditions determine which transition comes after this activity. The **Forward to Notify Client** activity does not specify any conditions by default.

You can add an error condition to this activity. The activity transitions through the error condition if there is an issue with the Notify client.

**Call workflow activity**

The **Call** activity makes outbound phone calls using a Notify workflow.

### Input variables

Input variables determine the initial behavior of the activity.

#### Table 1298: Input variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify Number</td>
<td>The Notify phone number to make the call from. When you initiate a call, the outgoing call workflow for the number group associated with this number runs.</td>
</tr>
<tr>
<td>Phone number to call</td>
<td>The E.164-compliant phone number to call.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use a script to determine number to call, and the Notify number to call from instead of using the <strong>Phone number to call</strong> and <strong>Notify Number</strong> variables.</td>
</tr>
<tr>
<td>Script</td>
<td>Define a script that controls which number to call. This script should return a string listing the Notify number sys_id, as well as the phone number to call, such as <code>{notify_number: 'sys_id', phone_number: '+316...'}</code></td>
</tr>
</tbody>
</table>

### Conditions

The conditions determine which transition comes after this activity. The **call** activity does not specify any conditions by default.
You can add an error condition to this activity. The activity transitions through the error condition if the call could not be set up due to invalid data returned by the advanced script.

*Join Conference Call workflow activity*

The **Join Conference Call** connects an incoming or outgoing call to a Notify conference call.

Notify includes the workflows Notify: (Re)join Conference Call and Notify: Join Conference Call Via SMS to demonstrate how to use the **join conference call** activity to connect inbound and outbound calls, and inbound SMS messages to a conference call.

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 1299: Input variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced</td>
<td>Select this check box to display advanced configuration options.</td>
</tr>
<tr>
<td>Script</td>
<td>Specify advanced configuration options using JavaScript, such as if the new participant should be muted upon joining the conference call. You can access values from the workflow scratchpad.</td>
</tr>
</tbody>
</table>

**Conditions**

The conditions determine which transition comes after this activity. The **join conference call** activity does not specify any conditions by default.

You can add an error condition to this activity. The activity transitions through the error condition if the conference_call scratchpad variable is not set.

**Scratchpad entries**

The activity uses the workflow scratchpad to read persistent values.
**Table 1300: Values read from scratchpad**

<table>
<thead>
<tr>
<th>Scratchpad variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow.scratchpad.conference_call</td>
<td>A GlideRecord for a single conference call record. A call processed by this activity is added to this conference call. If this value is not specified, the <a href="#">join conference call</a> activity will log an error. When initiating an outgoing call workflow using the Notify API <code>call(String notifyPhoneNumber, String toPhoneNumber, GlideRecord conferenceCall)</code> method, this scratchpad value is set automatically to the conference call GlideRecord. For incoming call workflows, or workflows initiated using a different mechanism, you must explicitly set this scratchpad value.</td>
</tr>
</tbody>
</table>

**Send SMS workflow activity**

The **Send SMS** activity sends short text messages to users’ phones using Notify.

**Input variables**

Input variables determine the initial behavior of the activity.

**Table 1301: Input variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Select the Notify phone number to use to send the SMS message.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to use JavaScript to determine which numbers to send the message to, and the Notify number to use to send the message.</td>
</tr>
<tr>
<td>To</td>
<td>Select any number of users to send the message to. The user record must have a E.164-compliant phone number or notification device configured for SMS messages.</td>
</tr>
<tr>
<td>To (groups)</td>
<td>Select any number of groups to send the message to. All members of that group with an E.164-compliant phone number or SMS notification device receive the message.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **Message** | Enter the message to send. You can add field values from the current record by using the **Select variables** box. You can also add values from the workflow scratchpad. If a field and a scratchpad variable have the same name, the field value is used.  
Because you can use variables in this message, it is not possible to determine the length of the message at design time. If the activity sends a message that is longer than supported by the telephony provider, the message is truncated and the instance logs a warning. |
| **To (script)** | Enter a script to determine which numbers to send the message to, and the Notify number to use to send the message.  
The script should return a JavaScript object with the format `{notify_number: '\...sys_id...\', users: [...], groups: [...], numbers: [...]}`.  
Specify the users or groups to send the message to as an array of sys_id values. Specify other numbers as an array of E.164-compliant phone numbers.  
This field appears only if **Advanced** is selected. |

**Conditions**

The conditions determine which transition comes after this activity. This activity does not specify any conditions by default.

You can add an error condition to this activity. The activity transitions through the error condition if the Notify number used to make the call is not configured correctly or unable to send SMS messages, or if an error occurs while sending the SMS.

*Queue workflow activity*

The **Queue** activity places an active Notify call in a queue.

Add the **Queue** activity to a workflow on the Notify Call [notify_call] table to put the current call on hold. This activity does not specify any input variables.

The queue that the call is added to is given a random ID.

**Notification workflow activities**

Notification workflow activities notify users of events that occur during the workflow.

*Create Event workflow activity*

The **Create Event** activity adds an event to the event queue, but does not immediately fire the event.

The event processor typically processes the event within one minute. This activity triggers any business rules or email notifications that would normally be triggered by the event.
Results

**Finished:** the activity added the event to the event queue.

Input variables

Table 1302: Create Event activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event name</td>
<td>The name of the event to create. From the Event Name lookup list, select the event to add to the queue. If the event requires parameters, specify them in the Parameter script field.</td>
</tr>
<tr>
<td>Parameter 1</td>
<td>The first parameter of the event.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If this parameter is a string value, it must be within quotes (&quot; &quot;).</td>
</tr>
<tr>
<td>Parameter 2</td>
<td>The second event parameter.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If this parameter is a string value, it must be within quotes (&quot; &quot;).</td>
</tr>
</tbody>
</table>

Notification workflow activity

The **Notification** activity sends an email or SMS message to specified users or groups.

Input variables

Table 1303: Notification activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>The users who will be recipients of the email.</td>
</tr>
<tr>
<td>To (groups)</td>
<td>The members of the groups that will be recipients of the email.</td>
</tr>
<tr>
<td>Advanced</td>
<td>If selected, the script in the <strong>To (script)</strong> field is called to specify additional recipients of the email.</td>
</tr>
<tr>
<td>To (script)</td>
<td>If <strong>Advanced</strong> is selected, this script is called and should set the variable <em>answer</em> to a comma-separated list of user or group sys_ids that you want to add as recipients of the email.</td>
</tr>
<tr>
<td>Message</td>
<td>The subject line of the email.</td>
</tr>
<tr>
<td>Subject</td>
<td></td>
</tr>
</tbody>
</table>
### States

The activity state tells the workflow engine what to do with the activity.

**Table 1304: Notification activity states**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the run function of the activity.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
</tbody>
</table>

### Timer workflow activities

Timer activities pause workflows for set periods of time.

**SLA Percentage Timer workflow activity**

The **SLA Percentage Timer** activity pauses the workflow for a duration equal to a percentage of an SLA.

A workflow must run on the Task SLA table to use this activity.

### Results

**Table 1305: SLA Percentage Timer activity results**

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>The activity successfully reached the specified duration</td>
</tr>
<tr>
<td>Cancelled</td>
<td>The activity or workflow was canceled before the timer reached the specified duration</td>
</tr>
</tbody>
</table>

### Input variables

Input variables determine the initial behavior of the activity.
Table 1306: SLA Percentage Timer activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage</td>
<td>The duration to pause the workflow for, as a percentage of the current SLA</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.

Table 1307: SLA Percentage Timer states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The activity is in this state very briefly while initializing, after which it immediately changes to <strong>Waiting</strong>.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine waits until the SLA reaches the specified percentage. The engine then transitions the workflow to the next activity.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Timer workflow activity**

The **Timer** activity pauses the workflow for a specified period of time.

This duration can be an absolute time value or a relative value based on a defined schedule. It is best to adjust the **Duration** so the workflow can progress in a timely manner. To pause a workflow indefinitely until a condition is met, see **wait for condition**.

**Results**

Table 1308: Timer activity results

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>The activity successfully reached the specified duration.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>The activity or workflow was canceled before the timer reached the specified duration.</td>
</tr>
</tbody>
</table>
## Input variables

Input variables determine the initial behavior of the activity.

### Table 1309: Timer activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer based on</td>
<td>How the timer duration is computed. The options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>A user specified duration</strong>: The duration is based on the <em>Duration</em> fields, such as days and hours.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A relative duration</strong>: The duration is based on the <em>Relative duration</em> (such as End of Next Business Day) and <em>Wait</em> fields.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A date/time or duration field</strong>: The duration is based on the <em>Field</em> value and the <em>Wait</em> field.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Script</strong>: The duration is based on a script that returns the number of seconds.</td>
</tr>
<tr>
<td>Duration</td>
<td>The specific number of days and hours to wait before proceeding to the next activity in the workflow. This field is available only when Timer based on is <strong>A user specified duration</strong>.</td>
</tr>
<tr>
<td>Relative duration</td>
<td>The general number and length of business days to wait before progressing to the next workflow activity. This field is available only when Timer based on is <strong>A relative duration</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>The date/time or duration field that contains the elapsed wait-time before moving to the next workflow activity. This field appears when the Timer based on is <strong>A date/time or duration field</strong>.</td>
</tr>
<tr>
<td>Script</td>
<td>The script that sets ‘answer’ to the number of seconds for the duration. This field is available only when Timer based on is <strong>Script</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Wait                  | An additional timer adjustment when Timer based on is A relative duration or A date/time or duration field. The options are:  
  • The full duration: No modification of the calculated duration.  
  • A % of the duration: The duration is adjusted by multiplying the number of seconds by the (Percentage / 100).  
  • Some time before: The duration is shortened by the Time before days and hours.  
  • Some time after: The duration is lengthened by the Time after days and hours. |
| Percentage            | The Wait percentage value when Timer based on is A relative duration or A date/time or duration field.                                                                                             |
| Time before           | The modifier time value when Wait is Some time before.                                                                                                                    |
| Time after            | The modifier time value when Wait is Some time after.                                                                                                                   |
| Schedule based on     | The basic schedule the timer uses to count working hours. If a schedule is specified, the duration will only be considered for times that are specified on the schedule. For example, if the duration is 2 hours and the workflow begins at 4:00pm on a schedule that is 8am - 5pm, then it ends at 9:00am the next day. The options are:  
  • This workflow's schedule: The schedule uses workflow context date, time, and an optional Time zone based on value.  
  • A specific schedule: The schedule uses a pre-defined Schedule and an optional Time zone based on value.  
  • A schedule field: The schedule uses a value from a table and an optional Time zone based on value. |
| Schedule              | The pre-defined Schedule from a list. This field available only when Schedule based on is A specific schedule.                                                               |
| Schedule field        | A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when Schedule based on is A schedule field. |
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time zone based on</td>
<td>The time zone for calculating the duration. The time zone may be based on:</td>
</tr>
<tr>
<td></td>
<td>• <strong>No time zone</strong>: Default. Workflow uses the GMT time zone.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A specific time zone</strong>: A predefined Time zone.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A time zone field</strong>: A Time zone field to track time duration from a field on the form.</td>
</tr>
<tr>
<td>Time zone</td>
<td>The predefined time zone. This field is available only when <strong>Time zone based on</strong> is <strong>A specific time zone</strong>.</td>
</tr>
<tr>
<td>Time zone field</td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when <strong>Time zone based on</strong> is <strong>A time zone field</strong>.</td>
</tr>
</tbody>
</table>

### States

The activity state tells the workflow engine what to do with the activity.

#### Table 1310: Timer activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The <strong>Timer</strong> activity is in this state very briefly while initializing, after which it immediately changes to <strong>Waiting</strong>.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine waits until the timer reaches the specified duration. The engine then transitions the workflow to the next activity.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

### Example

You can use a **Timer** activity to pause the workflow until the end of the business day.
Figure 899: Timer example

In this example, the script evaluates the time between the `now` variable and the `eod` variable. The `eod` variable is defined, in 24 hour time, as 4:00 PM. The script then sets the `answer` variable to the difference between these variables, in seconds, and logs a message.

```javascript
// get now and calc end of day (4:00pm)
var now = new Date();
var eod = new Date();
eod.setHours(16);
eod.setMinutes(0);
eod.setSeconds(0);
answer = (eod.getTime() - now.getTime()) / 1000;
```
Task workflow activities

Task activities create and modify workflow tasks.

Task activities are only available when the workflow runs on a table that extends Task.

*Add Worknote workflow activity*

The **Add Worknote** activity adds text to the Worknotes field of the current incident record.

A workflow must run on the Incident table to use this activity.

**Input variables**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Note</td>
<td>The text to add to the incident record.</td>
</tr>
</tbody>
</table>

*Attachment Note workflow activity*

The **Attachment Note** activity adds a comment to a journal field containing a link that points towards an attachment.

Attachments added to a record from this activity do not appear at the top of a record, but open a URL containing the content. This activity allows the use of irregular HTML tags to reference attachments, specifically the **[code]** tag. Entries in a journal field that use irregular HTML do not work if the glide.ui.allow_deep_html_validation property is true (It is set to false by default).

**Results**

- **Finished**: the activity added the attachment to the record.

**Input variables**

The following variables determine the behavior of the activity.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>A journal field to note the attachment. The options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>none</strong> (defaults to Work Notes)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Additional Comments</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Work notes</strong></td>
</tr>
<tr>
<td>Attachment Name</td>
<td>The name of the attachment in the URL. This name appears on the attachment at the top of the form.</td>
</tr>
</tbody>
</table>
### Catalog Task workflow activity

The **Catalog Task** activity creates a service catalog task record.

A user must complete the catalog task. This activity is available only for workflows running on the Catalog Request Item `[sc_req_item]` table.

### Results

You can assign a result value using `activity.result` from within a script field of the activity. The final **State** value of the catalog task record determines the result value for the **Create Task** activity. Possible result values are:

- Closed complete
- Closed incomplete
- Closed skipped
- Deleted
- Cancelled

### Input variables

The following variables determine the behavior of the activity.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>The Priority that is assigned to the task.</td>
</tr>
<tr>
<td>Wait for completion</td>
<td>If selected, the workflow activity waits for the task to complete before continuing. If cleared, the task is created but the workflow proceeds.</td>
</tr>
</tbody>
</table>
| Due date based on      | The due date fields are used to determine the values to use for setting **Expected Start Time** and **Due Date** for the task. The type determines how the due date is computed:  
  - **A user specified duration**: The duration is based on a user specified value.  
  - **A relative duration**: The duration is calculated from a relative duration (such as End of Next Business Day).  
  - **A date/time or duration field**: The duration is based on the value of a field on the current record.  
  - **Script**: The duration is returned by a script. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>The specific number of days and hours. This field is available only when <strong>Due date based on</strong> is <strong>A user specified duration</strong>.</td>
</tr>
<tr>
<td>Relative duration</td>
<td>The general number and length of business days. This field is available only when <strong>Due date based on</strong> is <strong>A relative duration</strong>.</td>
</tr>
<tr>
<td>Due date field</td>
<td>The date/time or duration field. This field appears when the <strong>Due date based on</strong> is <strong>A date/time or duration field</strong>.</td>
</tr>
<tr>
<td>Due date script</td>
<td>The script that sets 'answer' to the number of seconds for the duration. This field is available only when <strong>Due date based on</strong> is <strong>Script</strong>.</td>
</tr>
<tr>
<td>Schedule based on</td>
<td>The basic schedule the timer uses to count working hours. If a schedule is specified, the duration will only be considered for times that are specified on the schedule. For example, if the duration is 2 hours and the workflow begins at 4:00pm on a schedule that is 8am - 5pm, then it ends at 9:00am the next day. The options are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>This workflow's schedule</strong>: The schedule uses workflow context date, time, and an optional <strong>Time zone based on</strong> value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A specific schedule</strong>: The schedule uses a pre-defined <strong>Schedule</strong> and an optional <strong>Time zone based on</strong> value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A schedule field</strong>: The schedule uses a value from a table and an optional <strong>Time zone based on</strong> value.</td>
</tr>
<tr>
<td>Schedule</td>
<td>The predefined <strong>Schedule</strong> from a list. This field available only when <strong>Schedule based on</strong> is <strong>A specific schedule</strong>.</td>
</tr>
<tr>
<td>Schedule field</td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when <strong>Schedule based on</strong> is <strong>A schedule field</strong>.</td>
</tr>
<tr>
<td>Time zone based on</td>
<td>The time zone for calculating the duration. The time zone may be based on:</td>
</tr>
<tr>
<td></td>
<td>• <strong>No time zone</strong>: Default. Workflow uses the GMT time zone.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A specific time zone</strong>: A predefined <strong>Time zone</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>A time zone field</strong>: A <strong>Time zone field</strong> to track time duration from a field on the form.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Time zone</td>
<td>The predefined time zone. This field is available only when Time zone based on is A specific time zone.</td>
</tr>
<tr>
<td>Time zone field</td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when Time zone based on is A time zone field.</td>
</tr>
<tr>
<td>Task value from</td>
<td>The values used to create the task may either come from:</td>
</tr>
<tr>
<td></td>
<td>• Fields: a predefined set of fields including Fulfillment group, Assigned to, Short description and Instructions.</td>
</tr>
<tr>
<td></td>
<td>• Template: an existing template for the selected task table.</td>
</tr>
<tr>
<td></td>
<td>• Values: values that you specify using a Set Values widget.</td>
</tr>
<tr>
<td>Fulfillment group</td>
<td>The group that is responsible for completing the task.</td>
</tr>
<tr>
<td>Assigned to</td>
<td>The user that is responsible for completing the task.</td>
</tr>
<tr>
<td>Short description</td>
<td>A short description for the task.</td>
</tr>
<tr>
<td>Instructions</td>
<td>The task instructions for the user to complete prior to closing the task. These instructions appear in the Description field.</td>
</tr>
<tr>
<td>Advanced</td>
<td>If selected, the Advanced script is called to allow for additional task values to be set.</td>
</tr>
<tr>
<td>Advanced Script</td>
<td>Set additional values for the task in this script. Use the variable task when setting additional values. This script is run after the task values are set using the Fields, Template or Values you have specified. For example:</td>
</tr>
<tr>
<td></td>
<td>task.short_description = current.short_description;</td>
</tr>
<tr>
<td>Variables on Task Form</td>
<td>A slush-bucket of optional catalog variables to include. The variables here are displayed in the Catalog Task form in a field called Variable Editor. If no variables are defined in the field on the Catalog Task activity, the Variable Editor field in the Catalog Task form is not visible.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity.
### Table 1314: Catalog Task activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the <code>onExecute</code> function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

**Create Task workflow activity**

The **Create Task** activity generates a record on any of the tables that extend Task [task].

If the **Wait for completion** check box is selected, the workflow context waits for a user action on the task, such as Complete or Incomplete, and then progresses based on the user action.

**Note:** If a workflow contains a **Create Task** activity that has executed on the current record, additional **Create Task** activities in the workflow or in subflows cannot create new task records. To create additional tasks in this situation, you can add a **Run Script workflow activity** on page 3740 activity to the workflow that creates a task with a script.

### Results

You can assign a result value using `activity.result` from within a script field of the activity. By default, the final **State** value of the task record determines the result value for the **Create Task** activity. Possible result values are:

- Closed complete
- Closed incomplete
- Closed skipped
- Deleted
- Cancelled

### Input variables

The following variables determine the behavior of the activity.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Task Activity Settings</td>
<td>The following fields specify the behavior of the Create Task Activity.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Task type</td>
<td>The type of task to create. Select from the corresponding task table for the workflow.</td>
</tr>
<tr>
<td>Priority</td>
<td>The priority that is assigned to the task.</td>
</tr>
<tr>
<td>Wait for completion</td>
<td>If selected, the workflow activity waits for the task to complete before continuing. If cleared, the task is created but the workflow proceeds.</td>
</tr>
</tbody>
</table>

**Create Task Record Settings**

The following fields specify the field values that this activity sets for the task it creates.

<table>
<thead>
<tr>
<th>Task values from</th>
<th>The values used to create the task may either come from:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Fields</td>
<td>a predefined set of fields including <strong>Fulfillment group</strong>, <strong>Assigned to</strong>, <strong>Short description</strong> and <strong>Instructions</strong>.</td>
</tr>
<tr>
<td>- Template</td>
<td>an existing template for the selected task table.</td>
</tr>
<tr>
<td>- Values</td>
<td>values that you specify using a Set Values widget.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fulfillment group</th>
<th>The group that is responsible for completing the task.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigned to</td>
<td>The user that is responsible for completing the task.</td>
</tr>
<tr>
<td>Short description</td>
<td>A short description for the task.</td>
</tr>
<tr>
<td>Instructions</td>
<td>The task instructions for the user to complete prior to closing the task. These instructions appear in the <strong>Description</strong> field.</td>
</tr>
</tbody>
</table>

**Task template**

Only appears when **Task values from** is set to **Template**.

A template that is used to fill in values for the task.

**Set values**

Only appears when **Task values from** is set to **Values**.

A widget that is used to specify values for any fields of the task.

**Advanced**

If selected, the **Advanced script** is called to allow for additional task values to be set.

**Advanced Script**

Set additional values for the task in this script. Use the variable `task` when setting additional values. This script is run after the task values are set using the **Fields**, **Template** or **Values** you have specified. For example:

```plaintext
task.short_description = current.short_description;
```
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task Schedule</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Due date based on | The due date fields are used to determine the values to use for setting **Expected Start Time** and **Due Date** for the task. The type determines how the due date is computed:  
  - **A user specified duration**: The duration is based on a user specified value.  
  - **A relative duration**: The duration is calculated from a relative duration (such as End of Next Business Day).  
  - **A date/time or duration field**: The duration is based on the value of a field on the current record.  
  - **Script**: The duration is returned by a script. |
| Duration | The specific number of days and hours. This field is available only when **Due date based on** is **A user specified duration**. |
| Relative duration | The general number and length of business days. This field is available only when **Due date based on** is **A relative duration**. |
| Due date field | The date/time or duration field. This field appears when the **Due date based on** is **A date/time or duration field**. |
| Due date script | The script that sets 'answer' to the number of seconds for the duration. This field is available only when **Due date based on** is **Script**. |
| Schedule based on | The basic schedule the timer uses to count working hours. If a schedule is specified, the duration will only be considered for times that are specified on the schedule. For example, if the duration is 2 hours and the workflow begins at 4:00pm on a schedule that is 8am - 5pm, then it ends at 9:00am the next day. The options are:  
  - **This workflow's schedule**: The schedule uses workflow context date, time, and an optional **Time zone based on** value.  
  - **A specific schedule**: The schedule uses a pre-defined **Schedule** and an optional **Time zone based on** value.  
  - **A schedule field**: The schedule uses a value from a table and an optional **Time zone based on** value. |
| Schedule | The predefined **Schedule** from a list. This field available only when **Schedule based on** is **A specific schedule**. |
### Field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule field</td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when Schedule based on is A schedule field.</td>
</tr>
<tr>
<td>Time zone based on</td>
<td>The time zone for calculating the duration. The time zone may be based on:</td>
</tr>
<tr>
<td></td>
<td>• No time zone: Default. Workflow uses the GMT time zone.</td>
</tr>
<tr>
<td></td>
<td>• A specific time zone: A predefined Time zone.</td>
</tr>
<tr>
<td></td>
<td>• A time zone field: A Time zone field to track time duration from a field on the form.</td>
</tr>
<tr>
<td>Time zone</td>
<td>The predefined time zone. This field is available only when Time zone based on is A specific time zone.</td>
</tr>
<tr>
<td>Time zone field</td>
<td>A date and time or duration field for the schedule, that is associated with the table. Valid fields appear in blue on the Select the element from a tree dialog. This field is available only when Time zone based on is A time zone field.</td>
</tr>
</tbody>
</table>

### States

The activity state tells the workflow engine what to do with the activity.

**Table 1316: Create Task activity states**

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the onExecute function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is fired.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

### Utility workflow activities

Utility workflow activities provide controls over the path of the workflow and other useful tools.  

*Branch workflow activity*

The **Branch** activity splits the workflow into multiple transition paths from a single activity.
The Branch activity provides a single **Always** condition. You can draw any number of transitions from this condition. Using this activity is equivalent to drawing multiple transitions from a single condition of another activity. You can also add conditions to selectively transition to one or more activities.

**Note:** Branching can affect the behavior of rollback activities. See *Rollback To workflow activity* on page 3764 for more information.

**Join workflow activity**
The **Join** activity unites multiple execution paths into one transition.

Use this activity to cause a workflow to wait for all activities in multiple paths to finish before continuing. If multiple concurrent workflow paths meet without a **Join** activity, any subsequent activities execute twice.

To add Join to the canvas, click **Submit**. On the canvas, connect incoming transitions from each activity you want to act as a predecessor to the Join activity. Then connect outgoing transitions to the two exit conditions: Complete and Incomplete.

**Results**

Provide an Incomplete transition out of a **Join** whenever it is possible for any predecessor activities to follow a transition path that does not lead to the **Join** activity.

**Table 1317: Join activity results**

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td><strong>Join</strong> exits along the Complete path when the system has determined that all predecessor activities have completed and transitioned to the <strong>Join</strong>.</td>
</tr>
<tr>
<td>Incomplete</td>
<td><strong>Join</strong> exits along the Incomplete path when the system determines that at least one predecessor activity completed but transitioned along a path that bypassed the <strong>Join</strong> activity.</td>
</tr>
</tbody>
</table>

**Lock workflow activity**
The **Lock** activity prevents other instances of this workflow from continuing past this activity until the lock is released.

Several instances of the same workflow may run concurrently. For example, if a workflow is triggered when a record is added to a particular table and multiple records are added one after the other, that workflow will be triggered multiple times: once by each record insertion. In such cases, you can use the lock activity to ensure that this instance of the workflow has completely finished one or more activities before any other instance of the workflow can proceed.

A workflow can explicitly release a lock with the **Unlock** activity. The lock may also be released when the **Max duration** is reached.

When an instance of the workflow reaches the **Lock** activity, it attempts to obtain a lock using the key specified in the lock activity. If another instance has already obtained the lock and has not yet released it, this lock attempt fails. The instance continues trying to obtain the lock until **Max attempts** has been reached.

**Note:** We recommend placing a one-second timer activity before the lock activity. This helps prevent a rare condition in which the lock activity may not be able to distinguish one workflow instance from another. This condition can occur because the entity owning the lock is not
specific workflow instance, but rather the code-execution thread in which that instance is running. In most cases, each workflow instance runs on a different thread, but adding a timer activity ensures that this is the case.

Figure 900: Example of Lock Activity Preceded by Timer Activity
Results

Table 1318: Lock activity results

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>The activity successfully obtained a lock. This instance of the workflow can proceed past this point, but other instances cannot until the lock is released.</td>
</tr>
<tr>
<td>Failure</td>
<td>After attempting to obtain the lock Max attempts times, the activity could not obtain the lock.</td>
</tr>
</tbody>
</table>

Input variables

Input variables determine the initial behavior of the activity.

Table 1319: Lock activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>A unique mutex key. The Unlock activity uses this key to release the lock.</td>
</tr>
<tr>
<td>Duration</td>
<td>The maximum time the lock persists. The lock is released after it reaches this duration. Releasing a lock this way is equivalent to running the Unlock activity.</td>
</tr>
<tr>
<td>Max. duration</td>
<td>Specify how the activity behaves if the lock attempt is denied. If the final lock attempt fails, the activity state will be set to 'timeout' and the activity result will be set to 'failed'.</td>
</tr>
<tr>
<td>Lock attempts</td>
<td>Specify the maximum number of times the activity may attempt to obtain the lock.</td>
</tr>
<tr>
<td>Delay between attempts</td>
<td>The amount of time required after a failed lock attempt before another lock attempt is allowed.</td>
</tr>
</tbody>
</table>

States

The activity state tells the workflow engine what to do with the activity.

Table 1320: Lock activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiting</td>
<td>The workflow engine is waiting to obtain a lock.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity successfully obtained the lock.</td>
</tr>
</tbody>
</table>
Geneva    ServiceNow    ServiceNow Platform

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeout</td>
<td>The activity could not obtain a lock within the number of attempts specified by the Max. attempts input variable.</td>
</tr>
</tbody>
</table>

Log Message workflow activity
The Log Message activity writes a message to the workflow log.
Use this activity to add entries to the workflow's log for debugging or tracing purposes.

Input variables
Input variables determine the initial behavior of the activity.

Table 1321: Log Message activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message</td>
<td>The message to log. This variable can be a string or a JavaScript expression that evaluates to a string.</td>
</tr>
</tbody>
</table>

Log Trace Message workflow activity
The Log Trace Message activity writes a trace message to the workflow log.
The trace message includes the activity name, the event that invoked the workflow, and the table of the current record. There are no variables or conditions. To log other data, use the Log Message workflow activity on page 3738 activity.

REST Message workflow activity
The REST Message activity enables an administrator to override the REST endpoint or supply the variables configured in the REST Message module.
The REST Message activity executes a dead link REST function (POST, PUT, GET, or DELETE) on an endpoint using values defined in the function record.

Note: If you want to use a MID Server to send the REST message, the MID Server must be accessible by the instance and configured to use SSH.

Input variables

Table 1322: REST Message activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST Message</td>
<td>Name of the REST message to run. This is a reference field to the REST Message [sys_rest_message] table (System Web Services &gt; Outbound &gt; REST Message).</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REST Message Function</td>
<td>Function to call that is defined in a REST message function. This is a reference field to the HTTP Method [sys_rest_message_fn] table (System Web Services &gt; Outbound &gt; REST Message). Available functions are put, post, get, or delete. You can edit functions in the HTTP Methods related list in each REST Message record.</td>
</tr>
<tr>
<td>REST Endpoint</td>
<td>REST endpoint to use instead of the Endpoint defined in the HTTP Method record. Leave this field blank to use the defined endpoint in the REST Message Function record.</td>
</tr>
<tr>
<td>Variables</td>
<td>Values to use for variables defined in the HTTP Method record. Use the following format for the string: name1=value1, name2=value2, ... For example, use name=${nameValue}, id=${idValue} where name and id are function variables. If either the variable name or value contains a comma or equal sign, escape those characters with a backslash.</td>
</tr>
<tr>
<td>Use MID Server</td>
<td>Check box for using a MID Server to send the REST message. A MID Server might be necessary to reach an endpoint within a firewall or a subnetwork that is not visible from the instance. If this check box is selected, but no MID Server is defined in the MID Server field, the workflow automatically attempts to find a MID Server based on IP range and the REST capability.</td>
</tr>
<tr>
<td>MID Server</td>
<td>Name of the MID Server to use. This field is available when Use MID Server is selected. The workflow ignores this parameter if the use_midserver parameter is disabled.</td>
</tr>
<tr>
<td>Sensor Script</td>
<td>The script to execute after the request has been made and a response has been received. You can access the full response body from the activity.output variable.</td>
</tr>
</tbody>
</table>

**Return Value workflow activity**

The **Return Value** activity returns a value to a parent workflow, when run from a subflow.

This activity has no variables or conditions. For more information, see *Workflows used as subflows* on page 3552.

Use this activity within a subflow to store data that the parent flow can access. The **Return Value** activity adds the data from the subflow's **value** variable to the parent workflow's **scratchpad**.
Scratchpad entries

The activity uses the workflow scratchpad to read and write persistent values.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The activity writes the value from the subflow value variable to the parent scratchpad. The parent workflow activity that runs the subflow includes a Map return value to variable that defines where the parent workflow stores the returned data. This data can be scalar, a stringifiable JavaScript object, or an expression that evaluates to a stringifiable JavaScript object.</td>
</tr>
</tbody>
</table>

Run Script workflow activity

The Run Script activity runs a script you provide.

**Note:** All changes to current are automatically updated. You do not need to call current.update().

Input variables

Input variables determine the initial behavior of the activity.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Script</td>
<td>Script to execute.</td>
</tr>
</tbody>
</table>

Set Values workflow activity

The Set Values activity sets values on the current record.

The variables are loaded from the current record.

**Note:** Using the Set Values activity to set the Approval field on a task does not cancel pending approvals. To approve a task in a workflow, use the Approval Action activity.

SOAP Message workflow activity

The SOAP Message activity uses SOAP messages defined in the System Web Services plugin and can call the messages using a MID Server.

Your instance must have access to a MID Server configured to use SOAP to run this activity.
## Input variables

Table 1325: SOAP Message activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP Message</td>
<td>soap_message</td>
<td>The SOAP Message defined under the System Web Services plugin's Outbound SOAP Message [sys_soap_message] table. (System Web Services &gt; Outbound &gt; SOAP Message)</td>
</tr>
<tr>
<td>SOAP Message Function</td>
<td>soap_message_function</td>
<td>The function to call that is defined in the SOAP Message. Functions are listed in the SOAP Message Functions related list in each SOAP Message record.</td>
</tr>
<tr>
<td>SOAP Endpoint</td>
<td>soap_endpoint</td>
<td>Endpoint to use instead of the SOAP endpoint value in the SOAP Message Function record. Leave this field blank to use the defined endpoint in the SOAP Message Function record.</td>
</tr>
<tr>
<td>Variables</td>
<td>variables</td>
<td>Variables to substitute into the SOAP Envelope defined in the SOAP Message Function record. Use this format for the string: namel=value1, name2=value2, . . .</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If either the name or value contains a comma or equal sign, escape these characters with a backslash.</td>
</tr>
<tr>
<td>Use MID Server</td>
<td>use_midserver</td>
<td>Check box for using a MID Server to send the SOAP message. A MID Server might be necessary to reach an endpoint within a firewall or a sub-network that is not visible from the instance. If this check box is selected (true), but no MID Server is defined in the MID Server field, Workflow automatically attempts to find a MID Server.</td>
</tr>
<tr>
<td>Field</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MID Server</td>
<td>midserver</td>
<td>Name of the MID Server to use. This field appears when you select the <strong>Use MID Server</strong> check box. The workflow ignores this parameter if the <code>use_midserver</code> parameter is disabled.</td>
</tr>
<tr>
<td>Sensor Script</td>
<td>sensor_script</td>
<td>The script to execute after the request has been made and a response has been received. You can access the full XML response body from the <code>activity.output</code> object.</td>
</tr>
</tbody>
</table>

**Turnstile workflow activity**

The **Turnstile** activity limits how many times a workflow can pass through the same point.

Use this activity to prevent infinite loops. This activity is useful alongside the *Rollback To workflow activity* on page 3702 activity.

**Results**

You can assign a result value using the `activity.result` variable from within a script field of the activity. By default, the activity script evaluates if the activity should continue to iterate or stop.

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continue</td>
<td>The <code>Allowed iterations</code> value is greater than the number of times the workflow accessed this activity.</td>
</tr>
<tr>
<td>Cancel</td>
<td>The workflow accessed this activity more times than the <code>Allowed iterations</code> value.</td>
</tr>
</tbody>
</table>

**Input variables**

Input variables determine the initial behavior of the activity.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowed iterations</td>
<td>Number of times the workflow can pass through this activity before the turnstile ends the loop.</td>
</tr>
</tbody>
</table>
Conditions

The conditions determine which transition comes after this activity.

Table 1328: Turnstile activity conditions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continue</td>
<td>If the workflow has returned to this point an amount of times less than the allowed iteration.</td>
</tr>
<tr>
<td>Cancel</td>
<td>If the workflow has returned to this point an amount of times more than the allowed iteration.</td>
</tr>
</tbody>
</table>

States

The activity state tells the workflow engine what to do with the activity.

Table 1329: Turnstile activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing</td>
<td>The workflow engine knows to start the onExecute function of the activity.</td>
</tr>
<tr>
<td>Waiting</td>
<td>The workflow engine ignores the activity until a specific event to restart the activity is triggered.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity finished running. See the result value for the outcome of the activity.</td>
</tr>
<tr>
<td>Cancelled</td>
<td>This activity, or the workflow that contains this activity, was canceled.</td>
</tr>
<tr>
<td>Error</td>
<td>A JavaScript error occurred. Review the logs for error details.</td>
</tr>
</tbody>
</table>

Unlock workflow activity

The Unlock activity releases a lock that was previously placed by the Lock activity.

To release a lock, specify the same lock key that was specified in the Lock activity. If the Lock activity had a Duration specified, and that duration has already passed, the lock will already be released.

Input variables

Input variables determine the initial behavior of the activity.
Table 1330: Unlock activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock key</td>
<td>The Mutex key that releases the lock. This key must match the key specified by a Lock activity. For more information, see Lock activity.</td>
</tr>
</tbody>
</table>

States

The activity state tells the workflow engine what to do with the activity.

Table 1331: Unlock activity states

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finished</td>
<td>The activity successfully released the lock.</td>
</tr>
</tbody>
</table>

Subflow activities workflow activity

Subflow activities run and manage workflows from a parent workflow.

Parallel Flow Launcher workflow activity

The Parallel Flow Launcher activity launches multiple subflows in parallel.

Workflows running in parallel execute simultaneously and may complete in any order. The activity can launch a single subflow or multiple subflows as needed. You can manage the input values and values returned for each subflow.

Note: The Parallel Flow Launcher activity waits until all subflows are completed before proceeding. If any subflow does not finish, the activity waits indefinitely.

Note: Do not launch a large number of subflows with the Parallel Flow Launcher activity. If overused, this activity can overburden the instance. For launching very large numbers of subflows, consider putting the Parallel Flow Launcher inside a loop controlled by a turnstile or other conditional activity and having it do batches of no more than 500 at a time.

Activity variables

Activity variables determine the initial behavior of the activity.

Table 1332: Parallel Flow Launcher activity input variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique name for the activity.</td>
</tr>
<tr>
<td>Stage</td>
<td>The stage to display when the workflow reaches the activity.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Inputs</td>
<td>Inputs to the subflows to run. Specify an array of name: value pairs for each input defined in the workflow being launched. The name and data type of each input variable entered must match those used by the subflow that this activity launches. For a detailed example, see Parallel Flow Launcher Example.</td>
</tr>
<tr>
<td>Parallel Flow Launcher selection</td>
<td></td>
</tr>
<tr>
<td>Workflow</td>
<td>The workflow to run.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Check Advanced, to enter a script that uses a WorkflowCoordinator object to manage the subflows. When you check Advanced, a text box appears where you can enter a script that specifies a unique workflow for each set of input variables. See WorkflowCoordinator object on page 3747 for more information.</td>
</tr>
<tr>
<td>Parallel Flow Launcher iteration</td>
<td>Specify parameters to tune the performance of batched workflows.</td>
</tr>
<tr>
<td>Count</td>
<td>If not specified by an inputs set, Count determines the number of subflows executed in parallel from this activity. If the Advanced option is not selected, ensure that this field is populated with a valid number.</td>
</tr>
<tr>
<td>Max flows</td>
<td>The maximum number of workflows this activity can launch. If this value is a positive integer, it overrides the max parameter used in the WorkflowCoordinator constructor.</td>
</tr>
<tr>
<td>Max simultaneous</td>
<td>The maximum number of parallel workflows this activity can run at one time. If this value is a positive integer, it overrides the poolsize parameter used in the WorkflowCoordinator constructor.</td>
</tr>
<tr>
<td>Parallel Flow Launcher Process</td>
<td></td>
</tr>
<tr>
<td>Process flow complete</td>
<td>To specify a script that runs after each subflow completes, check Process flow complete. If you check this field, a text box labelled Flow complete appears, where you can enter the script to run.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Flow complete        | The script that runs each time a subflow finishes. This field is available when the **Process flow complete** option is selected.  

The same script functions and variables available in other workflow scripts, such as those in the Run Script activity, are available here. For more information, see **Completed subflow values in scripts** on page 3748.  

The variable “flow” is available to this script. It is an object that contains the following information about the flow that is finishing:  

- **flow.output (String)**: The value that the subflow returns to the parent if it executed a Return Value activity before ending.  
- **flow.index (Number)**: The zero-relative index of the subflow that finished.  
- **flow.contextId (String)**: The sys_id of the workflow context for the completed subflow.  
- **flow.inputs (Object)**: The inputs that were passed to the subflow when its context was created and started.  
- **flow.status (String)**: The final state of the context. This corresponds to the **state** column in the subflow context record, which has possible values of **executing**, **finished**, **cancelled**, or **faulted**. (Since the flow is completed, it cannot be **executing** at this point.) |

**Parallel Flow Launcher Split**

| Process finished | To specify a script that runs after all subflows have finished, check **Process finished**. If you check this field, a text box labelled **Finished Script** appears where you can enter the script to run. The ‘coordinator’ variable is made available to this script and is an object that allows access to any of the finished subflows using the getFlow(index) method. For example:  

```javascript
for (var i = 0; i < coordinator.getNumFlows(); i++)
    writeFlowResultsToTable( i, coordinator.getFlow(i) );
``` |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finished script</td>
<td>The script that runs after all subflows launched by the activity are complete. You can use variables that contain completed flow information in this script. The ‘coordinator’ variable is made available to this script and is an object that allows access to any of the finished subflows using the getFlow(index) method. For example:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>```javascript</td>
</tr>
<tr>
<td></td>
<td>for (var i = 0; i &lt; coordinator.getNumFlows(); i++)</td>
</tr>
<tr>
<td></td>
<td>writeFlowResultsToTable( i,</td>
</tr>
<tr>
<td></td>
<td>coordinator.getFlow(i) );</td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>This field is available when Process finished is selected.</td>
</tr>
</tbody>
</table>

**States**

The activity state tells the workflow engine what to do with the activity. To view an activity's state, point to the activity. A pop-up window shows the State and Result of the activity. If the activity is in an error state, the pop-up window provides a brief Fault Description.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiting</td>
<td>The activity is waiting for all subflows to finish. All subflows have started and some may have finished.</td>
</tr>
<tr>
<td>Finished</td>
<td>The activity successfully completed all of the subflows.</td>
</tr>
<tr>
<td>Error</td>
<td>The activity encountered an error.</td>
</tr>
</tbody>
</table>

**WorkflowCoordinator object**

A WorkflowCoordinator object specifies which subflows to run and the input variables to pass to those subflows.

When using a WorkflowCoordinator object, you can create a Parallel Flow Launcher activity that launches multiple subflows. When using the activity without a WorkflowCoordinator object, you can only launch a single subflow multiple times. Use one of the following methods to specify a WorkflowCoordinator object for the activity when the Advanced activity input variable is selected.

- Reference a workflow scratchpad variable that contains an existing WorkflowCoordinator object. To save a WorkflowCoordinator object to the scratchpad, call the save(variableName) function on the WorkflowCoordinator object. You can reference the object using the value passed in the variableName parameter. For example, you can create a WorkflowCoordinator object in a Run Script activity, save the object using `<object>.save('coord')`, and then call this object by entering coord in the Workflow activity variable of a subsequent Parallel Flow Launcher activity.
• Define the WorkflowCoordinator within the **Workflow** activity variable. Add the `javascript:` identifier at the beginning of the script. The *Parallel Flow Launcher example* on page 3748 shows how to use a WorkflowCoordinator object in this way.

• Create a factory class to define the WorkflowCoordinator object. The system does not provide a factory class for WorkflowCoordinator by default.

**Completed subflow values in scripts**

The *Parallel Flow Launcher* activity exposes additional variables you can use in scripts.

**Table 1334: Additional variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>coordinator</td>
<td>The WorkflowCoordinator used when running the subflows. You can use this variable in the <strong>Finished script</strong> to perform any final operations. Additionally, you can use the WorkflowCoordinator in a later workflow activity by passing the activity name or sys_id to the <code>WorkflowCoordinator.load('&lt;Activity&gt;')</code> function. For example, to load the WorkflowCoordinator object from a <em>Parallel Flow Launcher</em> activity called <em>Launch Subflows</em>, enter <code>var coord = WorkflowCoordinator.load('Launch Subflows');</code> in a later activity.</td>
</tr>
</tbody>
</table>
| flow | The subflow launched by the activity that completed most recently. You can use this variable in the **Flow complete** script to perform any post-processing operations on each subflow. To get a complete subflow from a coordinator object, use `var flow = coord.getFlow(i);` where `i` is the numeric index of the subflow based on the order it was launched. These values are available from the completed subflow:  
  • `index`: the numerical index of this subflow based on the order it was launched  
  • `workflow`: the sys_id or name, depending on which you passed to the WorkflowCoordinator constructor, of the workflow used for this subflow  
  • `inputs`: any input values provided to the launched subflow  
  • `status`: status of the subflow context  
  • `output`: the value returned by the subflow  
  • `contextid`: the sys_id of the workflow context for the subflow |

__Parallel Flow Launcher example__
This example shows how to use the **Parallel Flow Launcher** activity with an array of input values and with a WorkflowCoordinator object.

**Sample workflow**

This example shows a SQL-based web server with four application nodes. A single subflow runs to provision the database, and multiple parallel subflows each configure an application node. Finally, a separate set of parallel subflows configures the nodes to use a load balancer and sets up the server DNS.

![Flowchart](image)

**Figure 901: Parallel flow launcher business case**

**Provision the application nodes**

The first **Parallel Flow Launcher** activity launches the **Provision Node** subflow four times. The activity passes a unique IP address to each subflow from an array in the **Inputs** variable. The scripts defined in the **Flow complete** and **Finished script** variables write log messages regarding the status of the subflows.
Activity Properties: Parallel Flow Launcher

Name: Provision application nodes

Stage:

Inputs:

- [u_host: '192.168.0.10'],
- [u_host: '192.168.0.20'],
- [u_host: '192.168.0.30'],

Advanced:

Workflow: Provision Node

Count:

Max flows: 25

Max simultaneous: 5

Process flow complete:

Flow complete:

```javascript
// To access the flow that completed, use the 'flow' variable.
var flowOutput = flow.output;

workflow.debug("Provision Node Finished. Context: " + flow.contextId);
```

Process finished:

Finished script:

```javascript
// To access the set of completed workflows use the 'coordinator' variable.
var flowResult = coordinator.getFlow(1).output;

workflow.debug("All Nodes Provisioned");
```

Update
Add nodes to the load balancer

The second **Parallel Flow Launcher** activity uses WorkflowCoordinator objects to specify which subflows to run. The `coordinator` variable stores the completed flow information from the previous **Provision Nodes** activity. The script then retrieves the IP address and port for each node that was provisioned. The `coord2` WorkflowCoordinator object runs the Add Node to Load Balancer subflow once for each node, using the retrieved IP address and port information as input variables. Finally, the `coord2` WorkflowCoordinator object runs the **SetupDNS** subflow once to configure the load balancer.

```javascript
var coordinator = WorkflowCoordinator.load("Provision Nodes");
var coord2 = new WorkflowCoordinator(
    workflow: 'Add Node to Load Balancer'
);
for (var i = 0; i < coordinator.getNumFlows(); i++) {
    var ip = coordinator.getInput(i).ip;
    var port = coordinator.getOutput(i);
    coord2.add(
        ip: ip,
        port: port
    );
}
var loadBalancerIP = "10.0.20.10";
coord2.add( (ip: loadBalancerIP, hostname: "www.snow1.net", 'SetupDNS');
coord2;
```

![Figure 903: Specifying which subflows to run](image)

Subflows and the Create Task activity

If a workflow contains a **Create Task** activity that has executed on the current record, additional task activities in the workflow might not execute as expected. This can happen when the same subflow containing a **Create Task** activity runs more than once in a parent flow. When the subflow reruns and attempts to execute the **Create Task** activity again, the system reopens the first task activity instead and does not create an additional task.

---

**Note:** An alternative to creating duplicate subflows that use the **Create Task** activity is to add a **Run Script** activity to the workflow that creates a task with a script.
Figure 904: The same create task activity runs twice in a workflow

In this configuration, the workflow does not run the same subflow containing a Create Task activity more than once. This allows the workflow to create additional tasks.
Figure 905: Running different subflows containing the Create Task activity

View workflow activity descriptions

Tooltips are available for workflow activities to help you understand how to use each activity.

1. To view an activity description in the workflow editor, point to the icon for the activity in the palette.
To modify activity descriptions, navigate to **Workflow > Administration > Activity Definitions** and edit the **Description** field.

**Note:** To view more information about an activity, double-click the activity on the canvas and then click the help icon in the title bar of the **Activity Properties** window.

**Elements in workflow activity definitions**

Each activity can specify a number of elements that control the behavior of the activity or are controlled by the activity.

Not all activities specify all possible elements. See **Workflow activities** on page 3676 for links to the activities provided by default. Each activity description includes a detailed explanation of the specific elements offered by that activity.
Table 1335: Workflow activity elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results</td>
<td>The possible activity.result value. The activity result usually determines which condition the activity transitions through.</td>
</tr>
<tr>
<td>Scratchpad entries</td>
<td>Scratchpad variables the activity depends on to run, or variables the activity writes to the scratchpad.</td>
</tr>
<tr>
<td>Input variables</td>
<td>Values that control the behavior of the activity. Fields to set these values appear on the Activity Properties form when a new activity is added to a workflow. See Activity variables on page 3563 for more information.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Determines which transition the activity follows after completing. See Manage workflow activity conditions on page 3759 for more information.</td>
</tr>
<tr>
<td>States</td>
<td>Determines how the workflow handles the record being executed.</td>
</tr>
</tbody>
</table>

Add an activity to a workflow

Available activities are displayed in the Core, Packs, and Custom tabs in the Workflow Editor palette.

1. In the Workflow Editor, check out a workflow.
2. To add a workflow activity, drag it from the Core, Packs, or Custom tab to the canvas and drop it on a transition line in the workflow body.

   The transition turns blue when it is connected to the new activity. The designer adds the activity to the flow at that point and displays the property form for the new activity.

   If an activity is greyed out, approval engines are enabled for the table on which the workflow runs. To use the activity, turn approval engines off for the table.
Figure 906: Unavailable activities and approval engines

3. Create any additional conditions needed for the activity and ensure that all exits are connected.

4. Run the workflow validation tool prior to publishing to detect missing or disconnected transitions that can cause a workflow to hang.

**Note:** All activity descriptions have a Table value. If this value is Global, the activity is available for use with any workflow regardless of the table selected in the workflow properties. Activities that identify a specific table appear in the palette only if the table configured for the workflow matches or extends the table identified in the activity.
Manage transitions between workflow activities

Transitions define the processing path of the workflow, depending on conditions defined in each activity. All conditions in an activity must have a transition and all transitions must have a connection to another activity or to the End activity.

Note:
Run the workflow validation tool prior to publishing to detect missing or disconnected transitions that could cause a workflow to hang.

1. Add transitions to the workflow using either of these methods:
   - Drag and drop an activity directly onto a transition line to connect it to the adjacent activities. The transition line turns blue when the connection is made. The system updates the transitions automatically to reflect the new sequence.
     - Drag the activity to an open area in the canvas and create the transitions manually. Click the yellow square on the right side of the activity condition and drag a connector to the next task.
2. You can draw multiple transitions from the same activity condition if the activity executes concurrently.
3. To remove a transition, click to highlight it, and then press **Delete**.

### Custom activity transitions

Controls on the **Approval - User** activity enable an administrator to add additional workflow transitions to the activity other than the default transitions of **Approved** or **Rejected**.

Transitions defined using this method do not become a permanent feature of the **Approval - User** activity. After a new transition is configured, that transition must be applied manually to subsequent instances of the activity, where desired.

### Manage workflow activity conditions

Activities contain default conditions that determine which transitions are followed.

For example, the **Approval - User** activity has two conditions, **Approved** and **Rejected**.

---

**Figure 908: Example of activity conditions**
You can use a JavaScript condition check to create custom conditions on Core workflow activities. (Custom activities do not support this feature.)

1. Right-click the activity and select **Add Condition** from the context menu.
2. In the **New Workflow Condition** dialog box, fill in the fields as appropriate (see table).

Table 1336: New Workflow Conditions form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The label that is displayed on the workflow.</td>
</tr>
<tr>
<td>Activity</td>
<td>Name of the activity for which this condition is submitted. This value is populated automatically by the system.</td>
</tr>
<tr>
<td>Short description</td>
<td>Brief description of this condition.</td>
</tr>
<tr>
<td>Condition</td>
<td>A JavaScript condition check. The following variables are available:</td>
</tr>
<tr>
<td></td>
<td>• <strong>current</strong>: Current record that the workflow is running against.</td>
</tr>
<tr>
<td></td>
<td>• <strong>activity.result</strong>: Result value set by the activity upon completion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>activity</strong>: Workflow Executing Activity (wf_executing) record. Used for advanced condition checks.</td>
</tr>
<tr>
<td></td>
<td>• <strong>activity.vars</strong>: Variables associated with the Workflow Executing Activity record. Used for advanced condition checks.</td>
</tr>
<tr>
<td>Skip during generate</td>
<td>If selected, the <strong>Generate activity</strong> does not follow this transition to generate approvals or tasks.</td>
</tr>
</tbody>
</table>

3. Click **Submit**.
4. To change the order in which conditions appear on the workflow activity, right-click the activity and select **Reorder Conditions**.
   A dialog box appears, with a list of the available conditions.
5. Drag the conditions to a new position in the list.
6. Click OK.

**Activity result value**

The result value specified by an activity controls the condition through which the activity transitions. Use the result value as part of the Condition field in the activity. For example, if the Condition field of an Approval - User activity contains `activity.result == 'rejected'`, the activity transitions through that condition when a rejection is received from the approver. Result values are set in the Script field of the activity definition.

**Duplicate a workflow activity**

You can duplicate an activity used in a workflow, including all the configured properties.

1. Right-click the activity and select Copy Activity from the context menu.
   The system automatically duplicates the activity, but does not create transitions.
2. Double-click the copy and configure the properties appropriately.
3. Drag the activity to a location in the workflow.
4. Add transitions.
Using workflow approval activities and rolling back workflows

When you work with approvals, you need to understand how approval activities interact with approval engines, how to correct a skipped approval workflow activity, and how rollbacks work.

Approval workflow activities and approval engines

Approvals can be managed by approval activities or approval engines, but not both. Approval activities can be used if approval engines are not turned on for the table associated with the workflow.

Approvals can be managed by approval activities or approval engines, but not both. Trying to use both can cause a range of issues. Approval activities are not available (greyed out) in the palette if approval engines are used on the specified table. If you hover over a greyed out approval activity in the palette, a comment with more information is provided. For more information about approvals and approval engines, see Approvals on page 451.

![Figure 910: Unavailable approval activities]

To turn approval engines off for the table, navigate to the System Properties > Approval Engines and change the setting for the table to Turn engines off.
**Figure 911: Turn off approval engines**

Correct a skipped approval workflow activity

While a workflow is in an active context, an approval activity can skip to the next activity.
An approval activity might skip for the following reasons:

- The approval user or group is missing or invalid (for example, sys_id).
- The approval user or group became inactive after the approval record was created.
- The activity is a *dot-walked* field, such as `current.opened_by.department.manager`, and it has a missing or invalid approval user or group.
- The business rule on the table that is associated with the workflow is invalid.

To correct a missed approval activity:

1. Navigate to **Workflow** > **Live Workflows** > **All Contexts**.
2. Click the date and time in the **Started** column for the workflow that is incorrectly processing approval activities.
3. In **Related Links**, click **Show Workflow**.
4. Review the portion of the workflow that executed, and then do one or more of the following:
   - Verify that after approval, the workflow progressed to the next activity. If a workflow failed to progress, check the business rules. For more information, see *Debugging Business Rules* on page 3839.
   - Point to each processed approval activity to find activities where the **State** is **Finished** and **Result** is **Skipped**.
5. Navigate to **Workflow** > **Workflow Editor** and open the workflow.
6. Double-click the skipped approval activity.
7. Click **Users** or **Groups**.
8. Assign an active user or group for the approval activity. For more information, see *Workflow error handling* on page 3664.

**Rollback To workflow activity**

When conditions in a workflow triggers a **Rollback To** activity, the workflow moves processing backward to a specified activity in the workflow and resets certain activities that have already executed back to their original state. This is useful when handling an unexpected failure or as part of a programmed logical flow.

When an activity is reset during a workflow rollback, the following happens:

- Approvals are reset to **Not Requested**.
- Tasks are reset to either **Open** or **Pending**. A rollback workflow path cannot create new tasks.

Activities that perform external system operations, such as deleting a file or sending an email, are not rolled back. Only approval and task activity states are reset.

A workflow can contain a single rollback, multiple rollbacks, or nested rollbacks in more complex workflows. The **Rollback To** activity resets activities based on the actual workflow sequence (transition line attachments) of activities between itself and the transitioned to activity, rather than using the execution order to determine where processing should restart.

**Rollback To workflow activity behavior**

The **Rollback To** workflow activity transitions directly to the activity specified by the transition line arrow.

Use the **Rollback To** activity for all workflows that use multiple or nested rollbacks. **Rollback To** resets the targeted task (the direct transition) to **Open**. All tasks that have executed between the **Rollback To** activity and the targeted task (rolled back task) are set to **Pending**.

(0) Begin

(1) Log Message
(2) Task 1
(3) Task 3
(4) Task 2
(5) Timer
(6) Approval 1
(7) Roll back to first task
(8) Task 1
(9) Task 2
(10) Approval 1
(11) Log approval
(12) Join
(14) Send email
(15) End
The state of (3) Task 3 does not change, since this activity does not directly transition from the rollback target activity. To see what activities were rolled back, select the **Workflow Transition History** related list and look at the **Rolled back** column.

The **Rollback To** activity (7) updates the following activities:

- (8) Task 1: reset to **Open**
- (9) Task 2: reset to **Pending**
- (10) Approval 1: reset to **Not Yet Requested**
Using variables in Notify workflow activities

Certain Notify workflow activities allow you to use variables, such as those from the workflow scratchpad, to determine the activity behavior. Each activity supports a maximum of 20 variables. The following activities allow variable substitution:

Table 1337:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Say</td>
<td>Supports variable substitution in the Text field only.</td>
</tr>
<tr>
<td>Activity</td>
<td>Notes</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Input</td>
<td>Supports variable substitution in the Text field only.</td>
</tr>
<tr>
<td>Play</td>
<td>Supports variable substitution in the URL field only.</td>
</tr>
<tr>
<td>Forward call</td>
<td>Supports variable substitution in the Phone number field only.</td>
</tr>
</tbody>
</table>

**Scratchpad variables**

You can call variables from the workflow scratchpad or the activity scratchpad using the syntax `${variable_name}`. You do not need to include either workflow.scratchpad or activity.scratchpad before the variable name. For example, to use the variable `activity.scratchpad.langCode = 'en-US'`, call `${langCode}` within the activity. If the scratchpad does not contain the specified variable, the variable evaluates to an empty value.

You can get values from objects on the scratchpad using the format `${object.value}`. For example, you can get the name of a user object, such as `workflow.scratchpad.user = {name: 'john.smith'}` by calling `${user.name}`.

**The digit variable**

The **Input** activity exposes the `${digit}` variable. Use this variable in each condition presented by the activity. The number read to the user is determined automatically by each condition. The caller can press that number to cause the activity to transition through that condition.

**Use multiple timer activities in one workflow**

Workflow timer activities store data independently of each other in an activity-specific scratchpad.

Previously, all timer activities in a workflow accessed a single, shared scratchpad, which could lead to conflicts when adding multiple timer activities to one workflow.

Timer scratchpads entries hold these values:

- workflow.scratchpad.endTime
- workflow.scratchpad.realStartTime
- workflow.scratchpad.retroactiveSecsLeft

**Publish a custom workflow activity**

When a user creates a custom activity and saves or submits it, that activity appears in the Custom and Packs tabs of the designer palette, but is only visible to the user who created it.

When configuration is complete, the user clicks Publish, which makes the activity accessible to other users on the instance with the workflow_admin or activity_creator role. Published activities are available for upload to the ServiceNow Store, can be added to workflows, and can be edited by any user with the proper roles.

To edit a published activity, click Checkout. When an activity is checked out by a user, only that user can modify it. The fields of a checked out activity are read-only for all other users. When the checked out
activity has been modified successfully, the user publishes it again. The system adds a new version of this activity to the Custom tab in the workflow editor palette.

**Note:** Activities you create and publish are only visible in the Packs tab if they were created in the current application scope.

---

**Locked versions**

Problems can arise if an activity version is checked out by a user and not checked back in, for example, when the user is sick or leaves the company. An activity in this state cannot be checked out for update.

A user with the admin role can return a locked activity to a published state. The administrator opens the locked activity from the Custom tab of the workflow editor, selects the checked-out version, and selects **Force Checkout**, and then **Publish**.

---

**Timeline Visualizations**

A timeline visualization is a representation of an organization's activities over time.

Typically, timeline visualizations are useful for quickly assessing the impact of future strategic and operational activities such as change requests and projects. Timeline visualizations come in two varieties: a two-dimensional (2D) view where activities are grouped by month, and a three-dimensional (3D) view where activities are grouped in lanes. Both views are interactive, and the 2D view can be printed.

Timeline administrators, users with the timeline_admin role, can set up visualizations to represent ITSM-related activities, such as incidents, problems, changes, and projects. Timeline administrators or users with the timeline_user role can personalize their timeline visualizations from the Settings pane in the visualization.

Activating timeline visualizations adds the predefined CIO Roadmap timeline visualization, which shows your organization's projects grouped by portfolios. Organizational leaders can use the CIO Roadmap to monitor and evaluate the status of current and upcoming projects.

---

**Activating Timeline Visualization**

An administrator can activate the Timeline Visualization plugin to access the functionality. The Project Management plugin is activated if it is not already active.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
   - If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. Optional: If available, select the **Load demo data** check box.
   - Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.
Create a Timeline Visualization

Timeline administrators set up timeline visualizations for the organization’s leaders by creating a timeline that provides visual representations of the organization's operational and strategic activities. Additionally, they create timeline visualization views to define what data appears in the summary window when a user clicks a panel on the timeline.

- To create a timeline visualization, navigate to **Timeline Visualization > Create New** and create a new record (see table for field descriptions).
### Table 1338: Timeline visualization configuration form field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unique name for the visualization.</td>
</tr>
<tr>
<td>Panel table</td>
<td>Table that provides the records displayed as lanes and panels in the timeline.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The list shows only tables and database views that are in the same scope as the visualization.</td>
</tr>
<tr>
<td>Relationship field</td>
<td>Field on the table that contains values displayed as lane titles. Typically this field is a reference field or a field that contains a limited range of values.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The CIO Roadmap timeline visualization is a ServiceNow customized visualization for the Project application that does not use the Relationship field.</td>
</tr>
<tr>
<td>Show slider</td>
<td>Check box that enables (selected) or disables (cleared) displaying the timeline slider that users move to change the dates shown.</td>
</tr>
<tr>
<td>Panel name</td>
<td>Field from the Panel table that contains the values displayed in the panel body.</td>
</tr>
<tr>
<td>Panel date</td>
<td>Field from the Panel table that contains the date values displayed in the panel head in 3D view and in the panel body in 2D view. These dates also determine placement of panels on a lane. Only date and date-time fields are available on the choice list.</td>
</tr>
<tr>
<td>Default</td>
<td>Check box that sets (selected) or removes (cleared) the default status of a visualization when you have more than one defined for a specific table. Applications that include a visualization use the default visualization.</td>
</tr>
</tbody>
</table>
### Table 1339: Timeline visualization configuration filtering and sorting form field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane condition</td>
<td>Condition builder used to create filters and apply sorting to values that are used as lanes in 3D view visualizations. For example, if you set [Name] [is not] [IT] as a lane condition for the CIO Roadmap, then IT no longer appears as a lane in the roadmap, nor does it appear as a lane option in the Settings pane. Removing the filter restores the IT lane to the visualization and to lane options in the Settings pane. To order the results, specify sorting based on relevant field names. For example, to order the portfolio names so that they appear in reverse alphabetical order on the CIO Roadmap, set the sort fields to [Name] [z to a].</td>
</tr>
<tr>
<td>Panel condition</td>
<td>Condition builder used to create filters and apply sorting to values that are used as panels in 2D and 3D view visualizations. For example, if you set [State] [is one of] [Pending, Open, Work in Progress] as the panel condition for the CIO Roadmap, only projects that are in one of those states appear on the roadmap.</td>
</tr>
</tbody>
</table>

### Table 1340: Timeline visualization configuration color options form field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel color key</td>
<td>Field from the Panel table that contains values used for color coding the information displayed. The field selected here determines the values that are available in the Label fields on the form. The CIO Roadmap uses State, which is a field in the Project table. Panels on the CIO Roadmap are color coded according to the project state, which can be Pending, Open, Work in Progress, Closed Complete, Closed Incomplete, and Skipped. Examples of other fields that are suitable for this selection include Priority, Risk, and Approval.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Label 1</td>
<td>Values to be color coded. The values available are determined by the <strong>Panel color key</strong> field. For example, the CIO Roadmap is based on the Project table and has the <strong>Panel color key</strong> set to the <strong>State</strong> field, which contains the values Pending, Open, Work in Progress, Closed Complete, Closed Incomplete, and Closed Skipped. You can set specific colors for up to four values from the selected field. Other values are shown in the <strong>Default color</strong>.</td>
</tr>
<tr>
<td>Label 2</td>
<td></td>
</tr>
<tr>
<td>Label 3</td>
<td></td>
</tr>
<tr>
<td>Label 4</td>
<td></td>
</tr>
<tr>
<td>Default color</td>
<td>Color applied to values that are not selected for labels. For example, the CIO Roadmap color codes and creates labels for the values Pending, Open, Work in Progress, and Closed Complete. The additional values, Closed Incomplete and Closed Skipped, use the default color.</td>
</tr>
<tr>
<td>Color 1</td>
<td>Colors that correspond to each of the <strong>Label</strong> field values. For example, if <strong>Label 1</strong> is the Pending state, and <strong>Color 1</strong> is red, then panels for projects in the pending state are red.</td>
</tr>
<tr>
<td>Color 2</td>
<td></td>
</tr>
<tr>
<td>Color 3</td>
<td></td>
</tr>
<tr>
<td>Color 4</td>
<td></td>
</tr>
</tbody>
</table>

Create a Timeline Visualization view

The Timeline Visualization view determines which record details appear in the pop-up summary window when a user clicks a panel.

You can create one Timeline Visualization view per table. For example, if your view shows incident records, you can create a view on the Incident table to show only certain information, such as the incident number, priority, and short description.

If no Timeline Visualization view exists, all panels use the default view that shows all the fields that currently appear on the default view of the form for that table. After you create a Timeline Visualization view for a table, all panels then use that view.

For example, to create a Timeline Visualization view using the Incident form:

1. Navigate to **Incident > All**.
2. Open an incident.
3. Right-click the form header and select **Configure > Form Layout**.
4. Under **Form view and section**, go to the **View name** field and select **New**.
5. Enter **timeline visualization** in the **View name** field.
6. Click **OK** to create the timeline visualization view.
7. Add fields to or remove items from the timeline visualization view using the **Available** and **Selected** columns.
8. Click **Save**.
Use Timeline Visualization

Timeline Visualization provide a high-level view of an organization’s strategic and operational activities over time.

Organizational decision makers can use the information in visualizations for activities such as planning future projects and estimating resource requirements. Users can personalize the information displayed with the visualization Settings pane.

Activating the Timeline Visualization application also installs the CIO Roadmap, a timeline visualization that displays an organization's IT projects and portfolios. This following sections use the CIO Roadmap to demonstrate features of timeline visualizations.

Lanes

Lanes appear in 3D view only.

A visualization can display up to eight lanes at a time. While viewing a visualization, you can use the Settings pane to show or hide individual lanes.

Markers

Markers appear in 3D view only.

Markers are horizontal lines that cross all lanes and identify a transition to the next month.

Panels

A panel in 2D view always represents a single record, while panels in 3D view may represent one or more records.

Panels in both 2D and 3D views are color coded according to values that the administrator selects during the initial setup.

Figure 914: 3D panel

Figure 915: 2D panel

In 2D view, panels are grouped by month and stacked in chronological order, from the earliest date to the latest date. By default, the 2D view opens with the current month displayed on the left side of the visualization. You can print visualizations from the 2D view using the browser's print option. In 3D view,
panels are grouped in lanes and ordered by date, from earliest to latest. The date that appears on the panel determines its placement in 2D and 3D view. The date displayed is based on a value the timeline administrator selects during initial setup.

Panels appear in the CIO Roadmap according to the planned completion date for the project. In 3D view, projects with the same planned date of completion are consolidated into a single panel. In 2D view, projects with the same planned date of completion are displayed as individual panels.

Panel headers in the CIO Roadmap are color coded based on project state. However, in 3D view, if a panel represents more than one project, the panel header is colored black. The Settings pane contains a key showing each available project state and the corresponding color.

To view additional information about a panel:

• Click a panel for a single record while in 2D or 3D view to open a summary window that contains additional information. Click the heading in the summary window to open the full record.
• Click a panel that represents multiple records to open a list of those records. Click a record number to open the full record.

The timeline administrator can configure the information that appears in summary windows.

Personalize timeline visualizations

How to personalize timeline visualizations.

To personalize a timeline visualization, open the Settings pane and click Configure. Complete the form as appropriate (see table).
Table 1341: Timeline visualization personalization form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane conditions</td>
<td>* <strong>Condition builder</strong> used to create filters and apply sorting to values that are used as lanes in 3D view visualizations. For example, if you set [Name] [is not] [IT] as a lane condition for the CIO Roadmap, then IT no longer appears as a lane in the roadmap, nor does it appear as a lane option in the Settings pane. Removing the filter restores the IT lane to the visualization and to lane options in the Settings pane. To order the results, specify sorting based on relevant field names. For example, to order the portfolio names so that they appear in reverse alphabetical order on the CIO Roadmap, set the sort fields to [Name] [z to a].</td>
</tr>
</tbody>
</table>

Figure 916: Visualization personalization
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel conditions</td>
<td>Condition builder used to create filters and apply sorting to values that are used as panels in the visualization. For example, if you set [State] [is one of] [Pending, Open, Work in Progress] as the panel condition for the CIO Roadmap, only projects that are in one of those states appear on the roadmap.</td>
</tr>
</tbody>
</table>
| Panel color key     | Field from the Panel table that contains values used for color coding the information displayed. The field selected here determines the values that are available in the Label fields on the form.  

The CIO Roadmap uses State, which is a field in the Project table. Panels on the CIO Roadmap are color coded according to the project state, which can be Pending, Open, Work in Progress, Closed Complete, Closed Incomplete, and Skipped.  

Examples of other fields that are suitable for this selection include Priority, Risk, and Approval. |
| Label 1          | Values to be color coded. The values available are determined by the Panel color key field. For example, the CIO Roadmap is based on the Project table and has the Panel color key set to the State field, which contains the values Pending, Open, Work in Progress, Closed Complete, Closed Incomplete, and Closed Skipped.  

You can set specific colors for up to four values from the selected field. Other values are shown in the Default color. |
| Label 2          | |
| Label 3          | |
| Label 4          | |
| Default color    | Color applied to values that are not selected for labels. For example, the CIO Roadmap color codes and creates labels for the values Pending, Open, Work in Progress, and Closed Complete. The additional values, Closed Incomplete and Closed Skipped, use the default color. |
| Color 1          | Colors that correspond to each of the Label field values. For example, if Label 1 is the Pending state, and Color 1 is red, then panels for projects in the pending state are red. |
| Color 2          | |
| Color 3          | |
| Color 4          | |

**Note:**

If the **Max items per lane** field is set to more than 1000, you may observe a delay when displaying the timeline data using Internet Explorer (IE) as the browser.
Settings pane

The Settings pane appears in both the 2D and 3D views.

Element names in the Settings pane vary based on the table and fields used to create the visualization.

Figure 917: CIO roadmap settings pane

The Settings pane contains the following elements:
### Table 1342: Settings pane elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Identifies the type of information that is color coded on the timeline, such as state or priority. It also lists the color assigned to each possible value, such as pending state and open state.</td>
</tr>
<tr>
<td>Configure</td>
<td>Allows you to personalize a visualization by creating filters on lane information and panel information, specifying sort order for results, and reassigning panel colors. These changes affect your view of the visualization only.</td>
</tr>
<tr>
<td>View</td>
<td>Allows you to switch between 2D and 3D view.</td>
</tr>
<tr>
<td>View <code>&lt;table&gt;</code> List</td>
<td>Opens a separate browser tab showing the complete record list for the associated table.</td>
</tr>
<tr>
<td>Portfolio</td>
<td>Allows you to click lane names to add or remove them from the visualization. While the visualization is in 2D view, the Settings pane displays a Show all <code>&lt;records&gt;</code> button that allows you to override lane filters applied to the initial setup.</td>
</tr>
</tbody>
</table>

### Timeline visualization components

The components that appear in a timeline visualization depend on whether you are viewing it in 2D or 3D view.

Components can include **panels**, **lanes**, **markers**, and the **Settings pane**.
Figure 918: CIO roadmap 2D
Use the slider and slider track

The slider and slider track are useful for quickly navigating to any point in time on a visualization.

For the CIO Roadmap, the slider track allows you to quickly view the distribution of projects over time and determine project states. While in 3D view, click and drag the slider to move the timeline forward or backward. In 2D view, click and drag the slider track right or left to move the timeline forward or backward.

The slider track provides an alternative view of panels on a visualization. For the CIO Roadmap, a colored dot in the slider track indicates the presence of one or more projects for a given day. The different dot colors correspond to different project states. This allows you to scan the slider track to identify projects of interest by their color, and then use the slider or slider track to quickly navigate to those projects. For example, if you are interested in projects that are in a pending state, you can use the slider to navigate to the colored dots that represent that state.

---

**Note:** Dots in the slider track do not correspond one-to-one with panels in a visualization. A dot represents the presence of one or more projects on a given date. In 3D view, if there is more than one project for a given day, all of the projects for that day are combined into a single panel on the roadmap. When the slider track is populated, projects in a multi project panel are analyzed for project state and then represented as appropriately colored dots in the slider track.

Similarly, you can use the slider or slider track in conjunction with the calendar to quickly navigate to any point in time. For example, if you are concerned about resource issues in April 2014 because you see a
large cluster of dots during that time, dragging the slider or slider track to those dots brings those projects into view. Click the individual panels to view additional information.

View timeline visualization

How to view the CIO Roadmap timeline visualization.

1. Navigate to Project > Projects > All.
2. Scroll to the bottom of the list and click the Timeline Visualization related link.

By default, the CIO Roadmap opens in three-dimensional (3D) view with the current date at the forefront of the timeline. You can use the Settings pane to change between two-dimensional (2D) and 3D view. As soon as you view a timeline visualization, it is considered personalized. After that, you see your personalized view even if the timeline administrator makes changes to the default visualization.

Timeline administrators can create timeline visualizations on other tables. If you are not sure whether a timeline visualization exists for a certain application, use the application navigator to open the application of your choice, and click All. If a timeline visualization exists for the application's table, a Timeline Visualization related link appears.

Work with timeline visualizations

There are various tools that allow you to move visualizations backward and forward in time depending on whether you are in 2D or 3D view.

The following options are available for navigating between months.

Table 1343: 2D views

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dragging</td>
<td>Hold down the left mouse button and drag the slider track right or left to view other months.</td>
</tr>
<tr>
<td>Left and right arrow keys</td>
<td>Press the left or right arrow keys on the keyboard to move the timeline forward or backward.</td>
</tr>
<tr>
<td>Mouse wheel button</td>
<td>Scroll the wheel to move the timeline forward or backward.</td>
</tr>
<tr>
<td>Right and left navigation arrows</td>
<td>Click the arrows to move from month to month. Arrows do not appear if there are no additional months to view.</td>
</tr>
</tbody>
</table>

The following options are available for navigating stacked panels.

Table 1344: Navigating stacked panels

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dragging</td>
<td>Hold down the left mouse button and drag the stack up or down to view other panels.</td>
</tr>
</tbody>
</table>
The following options are available for navigating in 3D view.

Table 1345: 3D view

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up and down navigation arrows</td>
<td>Click the arrows in a stack to view additional panels for the same month. Arrows do not appear if there are no additional panels to view.</td>
</tr>
</tbody>
</table>

Timeline visualizations key terms

Key Terms

- Lane: A logical grouping of items on the 3D timeline visualization. For example, one lane might show pending projects while another lane shows open projects.
- Panel: A block that represents an item record in a 2D timeline visualization, or a block that represents one or more item records in a 3D timeline visualization.
- Summary view: A pop-up window that appears when a user clicks a panel in either the 2D or 3D timeline visualization. The summary view contains additional information about the record. Timeline administrators configure the information that appears in the summary view.
- Marker: A bar indicating the start of a month in a 3D timeline visualization.
- Slider: A tool used to control the time period seen in a 3D visualization.
- Slider track: An alternative view of the panels displayed in both the 2D and 3D timeline visualization.
Figure 920: CIO Roadmap 2d
Planning Your Timeline Visualization

Before creating a timeline visualization, you should have a clear idea of the information that you want to display.

This planning process helps you identify the required table and fields, and determine filtering and sorting requirements. Since users can personalize the visualization, avoid setting restrictive filters that may prevent users from finding categories that they are accustomed to seeing. You can use the CIO Roadmap as a starting point for creating your own timeline visualizations.

Timeline visualizations roles

The following roles can access timeline visualizations.

<table>
<thead>
<tr>
<th>Role Title [Name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeline administrator [timeline_admin]</td>
<td>Create and edit timeline visualization pages and view timeline visualizations.</td>
</tr>
</tbody>
</table>
### Timeline User

**Description:**
View timeline visualizations. Timeline user access is also granted to users with the `project_manager` and `project_user` roles.

<table>
<thead>
<tr>
<th>Role Title [Name]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeline user [timeline_user]</td>
<td>View timeline visualizations. Timeline user access is also granted to users with the <code>project_manager</code> and <code>project_user</code> roles.</td>
</tr>
</tbody>
</table>

---

## Installed with Timeline Visualizations

The following components are installed with timeline visualizations:

- Tables
- Plugins
- UI Policies
- Script Includes
- Client Scripts
- Business Rules

Demo data is available with timeline visualization.

### Tables

Timeline visualization adds the following table.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeline Visualization [roadmap_page]</td>
<td>Stores all available timeline visualizations.</td>
</tr>
<tr>
<td>Personalize Timeline [roadmap_user_page]</td>
<td>Stores timeline personalization settings for all timeline visualizations.</td>
</tr>
</tbody>
</table>

### Plugins

Timeline visualization activates the following plugin, if it is not already active.

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Plugin ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Management</td>
<td>com.snc.project_management_v3</td>
<td>Tools that aid in planning, organizing, and managing projects by applying basic task management processes.</td>
</tr>
</tbody>
</table>

### UI Policies

Timeline visualization adds the following UI policy.
### Script Includes

Timeline visualization adds the following script includes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roadmap2DUtil</td>
<td>Timeline Visualization - 2D API to generate data for lanes and panels.</td>
</tr>
<tr>
<td>RoadmapCommonUtil</td>
<td>Timeline visualization common utility to handle generic functions.</td>
</tr>
<tr>
<td>RoadmapConfig</td>
<td>Timeline visualization utility allowing configuration through the Timeline Visualization [roadmap_page] table.</td>
</tr>
<tr>
<td>RoadmapItems</td>
<td>Timeline visualization utility to get lane items.</td>
</tr>
<tr>
<td>RoadmapUtil</td>
<td>API to generate data for lanes and panels.</td>
</tr>
</tbody>
</table>

### Client Scripts

Timeline visualization adds the following client scripts.

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roadmap Color Choice</td>
<td>Timeline Visualization [roadmap_page]</td>
<td>Sets color values for the color choice fields in the timeline visualization configuration page.</td>
</tr>
<tr>
<td>Roadmap Item Table</td>
<td>Timeline Visualization [roadmap_page]</td>
<td>Allows changing the color fields for the item_color_key whenever there is a change to the panel table.</td>
</tr>
<tr>
<td>Roadmap On Load</td>
<td>Timeline Visualization [roadmap_page]</td>
<td>Loads the personalized version of the roadmap/visualization.</td>
</tr>
<tr>
<td>Roadmap Panel Table Change</td>
<td>Timeline Visualization [roadmap_page]</td>
<td>Allows changing the field name for item_color_key whenever there is a change to the panel/visualization table.</td>
</tr>
</tbody>
</table>
Business Rules

Timeline visualization adds the following business rule.

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check only one default</td>
<td>Timeline Visualization [roadmap_page]</td>
<td>Checks if another visualization is already set as default when user sets a new visualization as the default.</td>
</tr>
</tbody>
</table>

Self-service

The Self-Service application gives supported end users a clean, simple front end to their IT support organization.

By default, the Self-Service application provides access to common actions, such as viewing your homepage, accessing the Service Catalog, viewing knowledge articles, working with incidents, and taking surveys.

Service catalog requests

This page contains links to content on service catalog requests.

Add new request items

Use ServiceNow's web-based interface to add new items to a request.

Only an administrator or a user with the catalog_admin role can add a new item to a submitted request. Modifying a request does not resubmit it to the approval process.

1. Navigate to Service Catalog > Open Records > Requests and open an existing request.
2. Click Add New Item.
3. In the dialog box that appears, choose an item and a quantity to add.
4. Click OK.

Amazon EC2 instance request from the Service Catalog

Users can request the provisioning and deletion of virtual computers (Linux or Windows) in the Amazon EC2 virtual computing environment through the ServiceNow Service Catalog.

The Amazon EC2 offering requires the purchase and activation of ServiceNow Orchestration Automaton.

Request an instance

1. Navigate to Self-Service > Service Catalog.
2. Select Amazon EC2 Instance from the Request Application Infrastructure category.
3. Select the operating system, the type of instance (Large or Small), and the number of instances requested.
   See the specifications for Large and Small instances in Amazon EC2 Instance Types.
4. Type a reason for the request.
5. Click **Order Now** to order the instance.

The Order Status form appears, summarizing the request. A status bar tracks the progress of the approval and provisioning of the EC2 instance.
Upon successful creation of the instance, you receive an email containing the instance ID, IP address, and the public DNS for the instances created.

**Terminate an instance**

1. Navigate to Service Catalog > Routine Changes > Terminate Amazon EC2 Instance.
   
   A list appears displaying the instances that are assigned to you.
2. Select an instance from the list and click Order.
   
   This list does not support multi-line selection, and there is no undo feature.

**Check-out redirect property**

A property called One-step checkout redirect (glide.vm.checkout_redirect) controls the view presented to virtual machine requesters in the service catalog.

By default, this property is set to false, which redirects the view to the Order Status form when the requester clicks Order Now. When this property is set to true, ServiceNow redirects the requester to his or her My Virtual Assets portal. This property is located in Cloud Management > Administration > Properties.

**Delete request items**

Use ServiceNow's web-based interface to delete items from a request.

All users can delete items from their own requests any time prior to delivery. Modifying a request does not resubmit it to the approval process.

1. Navigate to Service Catalog > Open Records > Requests and open an existing request.
2. In the Requested Items related list, select the check box beside the item to delete.
3. In the Actions choice list, select Delete.
ServiceNow recalculates the price and delivery date for the order.

Place a request

Use ServiceNow's web-based interface for ordering predefined goods and services.

1. Navigate to **Self-Service > Service Catalog**.

   The default catalog view organizes items in categories and subcategories.

2. Select an item to order.

   **Note**: Some items are order guides, which combine related items for easier ordering. For example, an order guide called **New Employee Hire** presents a list of items that new employees typically receive (desk, phone, computer, email account). After you select the items you want to order, the order guide requests the information required and shows a separate time to delivery for each item ordered.

   The order screen requests any additional information that's needed, such as the requester's name or location and the date the item is needed. It also displays the item's delivery time after approval.
3. Enter complete and accurate information to expedite your order.
4. Order any other items required.
5. Click **Order Now** when the order is complete.
   ServiceNow displays a confirmation screen for your order:
You can view the order's progress in the **Stage** column.

### Note:
It may be useful to note the order number for future reference.

ServiceNow automatically notifies the approvers if approval is required and creates work orders to fulfill the order according to your organization's process. For example, if you order a new notebook computer, the purchasing group might receive a work order to order the computer, and the desktop services group might receive a work order to configure the new computer and deliver it to your office after it arrives.

### Self-Service HR portal

From the Self-Service HR Portal, you can get HR benefit and policy information, and request help from the HR team.
Figure 922: HR Portal
From the HR Portal, you can take the following actions.

- Search for HR knowledge articles that answer your questions. As you type, articles are suggested based on keywords.
- Open your HR profile from the link under My Stuff. You can update fields that are not disabled in your HR profile.
- View your HR requests, approvals (if you are an approver), and organization chart.
- Submit HR questions and requests that are not answered in HR knowledge articles.
- Open a chat window with an HR specialist. An administrator must enable the HR chat queue for the link to appear.
- Send an email to HR to create your request.

Note: The HR Portal page does not work in Microsoft Internet Explorer 8. Use the HR Catalog module to submit an HR case.

Submit a request from the HR Portal

You can open the HR Portal to submit questions and request help from the HR team.

There are three types of HR requests in the Get Help list on the HR Portal.

**Ask a benefit question**
Use this link when you have a question about your organization's benefits or policies. There are a number of predefined topics, such as 401K or retirement, or you can ask a general question.

**Make a request**
Use this link when action is needed, such as changing your HR profile information or requesting a leave of absence.

**Change employee status**
Use this link for onboarding new employees or offboarding employees who are leaving the organization. Typically, HR managers request employee status changes. Employees who do not have authority to create employee status change requests see a blank page.

1. Navigate to Self-Service > HR Portal.
2. Under Get Help, select the type of request to submit.
3. Select the item to request.
   The Short description is filled out, and a list of related knowledge articles based on keywords in the text appears below the text field where you enter additional information, such as your question.
4. Review the knowledge articles that appear, and if your question or request is resolved, exit the form. Otherwise, continue with the remaining steps.
5. Answer the questions in the request form.
   Questions vary depending on the item selected. Mandatory fields have a red asterisk (*) beside them.
6. Click Submit.
   The page refreshes to display your open HR cases. A link to the case appears at the top of the page, and, if the case was automatically assigned, a message tells you who the case was assigned to.

Submit a case from the HR catalog

You can submit HR cases from the HR catalog.

1. Navigate to Self-Service > HR Catalog.
2. Select the appropriate category and item.
3. Answer the questions in the request form.
   Questions vary depending on the item selected. Mandatory fields have a red asterisk (*) beside them.

4. Click **Submit**.
   The case is submitted to HR for processing. A link to the case appears at the top of the page, and, if
   the case was automatically assigned, a message tells you who the case was assigned to.

View your open HR cases

You can view your open HR cases from the HR Portal.

1. Navigate to **Self-Service > HR Portal**.
   You can also navigate to **Self-Service > My Requests**, however, the list that appears contains all of
   your open requests, not just HR cases.

2. Click **My open cases** under **My Stuff**.

3. Click the case number to see its details.
   Based on keywords in the **Short description** field, a list of knowledge articles may appear below the
   activity section. Review the articles for a resolution to your question or request.

4. Optional: Enter comments or change the priority if necessary.

5. Optional: To view your manager in a hierarchy, scroll to **Related Links** and click **Show employee org chart**.
   Click the back arrow on the upper left to return to the case.

6. If you made changes, click **Update**.

View request status

Use ServiceNow's web-based interface to view the status of a request.

1. Navigate to **Self-Service > Requested Items**.

2. Click your order number to view details.

3. View the order's progress in the **Stage** field.

Customize homepages

If you are assigned at least one role in the system, you can create a customized homepage that you can
see when you log in.

The content on your homepage comes from a variety of sources, such as graphs and charts generated
from a report, application modules, a service catalog category, the scrolling news widget, which is the
News knowledge category.

1. Navigate to **Self-Service > Homepage**.

2. Add any item to the homepage by clicking the add content icon ( ) or the **Add content** link at
   the top.
   A window will open that lets you select from a number of different homepage items you can add.
   Some items pertain to specific features and applications, such as CMS content blocks and the Work
   Management dispatch map. So the list of items varies depending on what is active on your system.

3. Reposition elements on your homepage by dragging and dropping them to a new location.

4. To remove a homepage item, click the [X] on the right side of the item's header.
Delete custom homepages

You can delete any of your custom homepages if you no longer need them.

1. Click the **Delete page** link at the bottom of the homepage.

2. Click **Ok** to confirm the deletion.

How Help the Help Desk identifies devices

Help the Help Desk uses a predefined series of queries to identify and update existing CIs in the CMDB or to create a new CI if no match is found.

These queries attempt to match devices using the three criteria listed here, in this order. Updates to an existing CI require only a single match as the list is evaluated. For example, if a device's name has changed, but the MAC address is the same, the CI with the matching MAC address is updated.

- Serial number in the cmdb_ci_computer table#
- MAC address in the cmdb_ci_network_adapter table
- Computer name in the cmdb_ci_computer table

**Note:** Discovery Identifiers are incompatible with Help the Help Desk queries.

Script Include

The script include **CIIdentifierForHelpDesk** provides the logic for updating existing CIs or creating a new CI if no matching device exists in the CMDB. *Do not* modify this script. Errors introduced into this script can result in update failures or in new CIs being created for every device found.

Running Help the Help Desk

Help the Help Desk is a tool that enables users to populate the CMDB automatically with information about their Windows computer. Help the Help Desk is a small Web application that downloads and runs locally, using a login script to gather information such as serial number, computer name, disk configuration, network configuration, installed software, memory, and much more.
To detect all system software successfully on a 64-bit machine, make sure to run the Help the Help Desk script from a 64-bit browser. A 64-bit browser can detect both 64-bit and 32-bit software, but a 32-bit browser cannot detect 64-bit software.

1. On your instance, navigate to **Self Service > Help the Help Desk**.
2. Click **Start the Scan to Help the Help Desk**.
   
   You are prompted to run or save the `discovery.hta` script.
3. If your browser is Internet Explorer, run the script. If you are using any other browser, click **Save** and save the script to the local machine.
4. To execute the saved script, double click the file.
   
   The script runs a series of WMI queries to gather information about the Windows machine. When it is done, the data is sent back to your ServiceNow instance and used to populate the configuration database (CMDB).

### Service Delegation

Service delegation is the ability to designate other users to view and interact with approvals and tasks assigned or sent to you, and to receive copies of all email notifications sent to you.

**Note:** Users to whom you delegate responsibility see all pending tasks and approvals regardless of the delegation period.

**Note:** Delegation does not cascade. When a user acts as a delegate for another user, the delegate cannot in turn delegate to a third party. The original delegate is the only user who receives assignments and notifications.

### Add the Delegates Related List to a user profile

By default, the delegates related list is hidden from the User Profile form. An administrator or a user with the personalize_form role must add the delegates related list to the User Profile form.

1. Navigate to **Self Service > My Profile**.
2. Configure the form and add the **Delegates->User** related list.

<table>
<thead>
<tr>
<th>Delegate related list</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delegates-&gt;Delegates</td>
<td>Shows a list of users for whom the current user is a delegate. For example, if the current user is the ITIL User, the list shows users where Delegate = ITIL User.</td>
</tr>
<tr>
<td>Delegates-&gt;User</td>
<td>Shows a list of users who are delegates of the current user. For example, if the current user is the ITIL Admin, the list shows delegates where User = ITIL Admin.</td>
</tr>
</tbody>
</table>

### Delegate approvals and tasks to another user

If a user is out of the office, s/he can delegate responsibilities to other users for a period of time.

1. Navigate to **Self Service > My Profile**.
2. In the **Delegates** related list, click **New**.

3. Select the Delegate (User).
4. Set the period of time.
5. Specify the responsibilities the delegate will assume using the check boxes. You can delegate:
   - Approvals: The delegate can approve items on your behalf.
   - Assignments: The delegate can view and work on tasks assigned to you.
   - CC Notifications: The delegate receives a copy of email notifications sent to you, except those marked Meeting Invitation.
   - Meeting Invitations: The delegate receives a copy of email notifications sent of the type Meeting Invitation.

If the list is not visible, configure the form to add the **Delegate->User** related list.
Note: You must select an active user as your delegate. The instance only delegates to active users.

A sample delegation looks like this:

![Delegate view](image)

Note: If your delegate gets email notifications, they will be the same email notifications sent to you. The delegate may be confused to see "Incident assigned to you" emails, so make sure they know they are a delegate.

Scripts

Use scripts to extend your instance beyond standard configurations. With scripts, you may automate processes, add functionality, integrate your instance with an outside application and more.

APIs (Application Programming Interfaces) provide classes and methods that you can use in scripts to define functionality. ServiceNow provides APIs as JavaScript classes, web services, and other points of connection for integrations. Note that you cannot access commonly used JavaScript objects (such as DOM or Window). Jelly scripts are also used in some modules. Jelly is used to turn XML into HTML and may include both client-side and server-side scripts.

Scripts may be server-side (run on the server or database), client-side (run in the user's browser) or on the MID server.

A variety of classes are available for use in scripts. The classes are grouped by those used for Client scripts on page 3900, REST APIs, global server scripts, and scoped server scripts. For detailed information on the classes and methods available, see the developer portal.

Note: When you are writing scripts, you cannot use reserved words.

It is recommended that you be familiar with JavaScript coding before you begin customizing your instance, and with Jelly if you intend to deploy Jelly scripts. The following topics provide general information, procedures, and contexts for scripting in the platform.

- Glide class overview on page 3953
- Syntax editor on page 3816
- Execution order of scripts and engines on page 470
When developing scripts for scoped applications, you must use the scoped APIs, which include scoped versions of the Glide APIs. The scoped Glide APIs do not provide all the methods included in the global Glide APIs, and you cannot call a global Glide API in a scoped application.

Script functions

Scripts can be used in many places. The most important detail is whether the script runs on the client or the server.

<table>
<thead>
<tr>
<th>Script</th>
<th>Function</th>
<th>Runs on</th>
</tr>
</thead>
</table>
| Access Control | Determines whether access will be granted for a specified operation to a specific entity.  
                  | - type of entity being secured  
                  | - operation being secured  
                  | - unique identifier describing the object  
                  | Can be defined by roles, conditional expressions or scripts.  | server - script and any condition run on the server |
| Ajax Scripts  | Enables the client to get data from the server to dynamically incorporate into a page without reloading the whole page.  
                  | - Ajax Client Scripts request that information be returned, or that action be taken, or sometimes both  
                  | - Ajax Server Scripts fulfill Ajax Client Script requests  | client - Ajax Client Scripts run on the client  
<pre><code>              |                                                              | server - Ajax Server Scripts run on the server |
</code></pre>
<table>
<thead>
<tr>
<th>Script</th>
<th>Function</th>
<th>Runs on</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Rules</strong></td>
<td>Customizes system behavior</td>
<td>server - script and any condition run on the server</td>
</tr>
<tr>
<td></td>
<td>• runs when a database action occurs (query, insert, update or delete)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• the script may run</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• before or after the database action is performed (runs as part of the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>database operation)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• asynchronously (at some point after the database operation)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• on display (when displaying the data in a form)</td>
<td></td>
</tr>
<tr>
<td>Catalog UI Policies</td>
<td>Defines the display of a variable set or a catalog item (from the</td>
<td>• client - scripts in the “execute if true” field or “execute if false” field run on the client</td>
</tr>
<tr>
<td></td>
<td>service catalog).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• server - all conditions run on the server</td>
</tr>
<tr>
<td><strong>Client script types</strong></td>
<td>Used for making changes to the appearance of forms, displaying different</td>
<td>client</td>
</tr>
<tr>
<td>on page 3901</td>
<td>fields based on values that are entered or other custom display options.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• onLoad means the Client Script runs when the form or page is loaded</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• onChange means the Client Script runs when something specific gets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>changed AND also when the form or page loads</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• onSubmit means the Client Script runs when the form is submitted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Client Scripts can also be called by other scripts or modules,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>including UI policies.</td>
<td></td>
</tr>
<tr>
<td>Script</td>
<td>Function</td>
<td>Runs on</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Script Actions</td>
<td>Contains scripts which run when an event occurs, for example</td>
<td>server - script and any condition run on the server</td>
</tr>
<tr>
<td></td>
<td>• approval is cancelled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• change is approved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• problem is assigned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can have a condition which must be true for the script to run. Commonly used to call a Script Include.</td>
<td></td>
</tr>
<tr>
<td>Script Includes</td>
<td>Contains scripts which can be functions or classes. These scripts run only when called by other scripts (often Business Rules).</td>
<td>server</td>
</tr>
<tr>
<td></td>
<td>Any server script which is complicated or reusable should be a Script Include (especially complicated Business Rules).</td>
<td></td>
</tr>
<tr>
<td>Transform Maps</td>
<td>Used for importing data.</td>
<td>server</td>
</tr>
<tr>
<td></td>
<td>• defines mapping relationships between tables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• can use Business Rules, other scripts and/or other options to import that data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Do not always include scripts.</td>
<td></td>
</tr>
<tr>
<td>UI Actions</td>
<td>Creates the ability to choose a specific action such as clicking a button or a link.</td>
<td>client - when the &quot;Client&quot; box is checked, the script in the script field runs on the client</td>
</tr>
<tr>
<td></td>
<td>UI Actions put these items on forms and lists:</td>
<td>server - when the &quot;Client&quot; box is unchecked, the script in the script field runs on the server</td>
</tr>
<tr>
<td></td>
<td>• buttons</td>
<td>client - when the &quot;Client&quot; box is checked, the onClick script is available, which can contain any JavaScript but normally calls a function which is specified in the script field</td>
</tr>
<tr>
<td></td>
<td>• links</td>
<td>server - all conditions run on the server</td>
</tr>
<tr>
<td></td>
<td>• context menu items</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• list choices</td>
<td></td>
</tr>
<tr>
<td>Script</td>
<td>Function</td>
<td>Runs on</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| UI Context Menus    | Defines which "right-click menu" will pop-up in which area, and the menu choices that will be available | • client - onShow scripts run on the client  
• client - action scripts run on the client  
• server - dynamic action scripts run on the server  
• server - all conditions run on the server |

**Note:** If you use a left-handed mouse configuration, right-click means "click the other button."

| UI Macros           | Contains modular, reusable components that can contain Jelly and are called by UI pages. They also contain different types of scripts and may be called multiple times on the same page. | • server - the UI Macro itself executes on the Server  
• server - may contain content that runs on the server (Jelly expressions or JavaScript inside Jelly constructs)  
• client - may generate output that runs on the client (embedded JavaScript within <script> tags) |
<table>
<thead>
<tr>
<th>Script</th>
<th>Function</th>
<th>Runs on</th>
</tr>
</thead>
</table>
| **UI Pages** | Used to create and display pages, forms, dialogs, lists and other UI components. Can be displayed on a standalone basis, or called as a usable component, as part of a larger page. | • server - Jelly XML runs on the server to produce HTML  
• client - HTML may contain embedded JavaScript that runs on the client  
• client - client scripts run on the client  
• server - processing scripts run on the server |

Can contain
• Client Scripts,
• processing scripts (which are server scripts),
• HTML,
• Jelly,
• UI Macros,
• and also can call other scripts.

**Note:** Jelly turns XML into HTML.
<table>
<thead>
<tr>
<th>Script</th>
<th>Function</th>
<th>Runs on</th>
</tr>
</thead>
</table>
| UI Policies | Defines the behavior and visibility of fields on a form.                 | • client - scripts in the "execute if true" field or "execute if false" field run on the client  
• server - all conditions run on the server |
|             | • mandatory                                                               |                                                                         |
|             | • visible                                                                 |                                                                         |
|             | • read only                                                               |                                                                         |
|             | Use UI Policies rather than client scripts whenever possible.             |                                                                         |
|             | • UI Policies are always attached to one table                            |                                                                         |
|             | • UI Policies often have a condition which must be true in order for them to run |                                                                         |
| UI Properties | Designates what the instance will look like.                           | • server - properties set on the server  
• client - the results get rendered on the client  
no scripts |
| UI Scripts | Contains client scripts stored for re-use. Only used when called from other scripts.  
Not recommended for use. | client |
| Validation Scripts | Validates that values are in a specified format.                      | client |
|             | For example, a validation script can verify that the only value allowed in a specific field is an integer. | |
| Workflow Editor | Used to create or change a workflow. Scripts can be run at any point in a workflow, or different scripts can be run at different points.  
Scripts also can be found inside every workflow activity and can be modified (although do so with extreme caution). | server - script and any conditions run on the server |
Execution order of scripts and engines

Scripts, assignment rules, escalations, and engines all take effect in relation to a database operation, such as insert or update. In many cases, the order of these events is important.

**Note:** Client-based code that executes in the browser, using Ajax or running as Javascript, will always execute before the form submission to the server.

The order of execution is as follows:

1. _Before_ business rules: Scripts configured to execute before the database operation with an order less than 1000.
2. _Before_ engines. The following are not executed in any specific order:
   - Approval engine (for task and sys_approval_approver tables)
   - Assignment rules engine (for task tables)
   - Data policy engine
   - Escalation engine
   - Field normalization engine
   - Role engine - keeps role changes in sync with sys_user_has_role table (for sys_user, sys_user_group, sys_user_grmember, and sys_user_role tables)
   - Execution plan engine (for task tables)
   - Update version engine - creates version entry when sys_update_xml entry is written (for sys_update_xml table)
   - Workflow engine (for default workflows)
3. _Before_ business rules: Scripts configured to execute before the database operation with an order greater than or equal to 1000.
4. The data base operation (insert, update, delete).
5. _After_ business rules: Scripts configured to execute after the database operation with an order less than 1000.
6. _After_ engines. The following are not executed in any specific order:
   - Label engine
   - Listener engine
   - Table notifications engine
   - Role engine - keeps role changes in sync with sys_user_has_role table (for sys_user, sys_user_group, sys_user_grmember and sys_user_role tables)
   - Text indexing engine
   - Update sync engine
   - Data lookup engine inserts or updates
   - Workflow engine (for deferred workflows)
7. Email notifications. The following are executed based on the weight of the notification record:
   - Notifications sent on an insert, update, or delete
   - Event-based notifications
8. _After_ business rules. Scripts configured to execute after the database operation with an order greater than or equal to 1000.
Scripting of field types

Table 1348: Scripting of field types

<table>
<thead>
<tr>
<th>Type</th>
<th>Evaluates to in script</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The string</td>
<td>&quot;dog&quot; &gt; &quot;dog&quot;</td>
</tr>
<tr>
<td>Decimal</td>
<td>A number with up to two decimal points</td>
<td>12.34 &gt; 12.34</td>
</tr>
<tr>
<td>Integer</td>
<td>A number with zero decimal points</td>
<td>12 &gt; 12</td>
</tr>
<tr>
<td>True / False</td>
<td>true or false</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>A date formatted as yyyy-mm-dd</td>
<td>2008-11-04</td>
</tr>
<tr>
<td>Date-time</td>
<td>A day and time formatted as yyyy-mm-dd hh:mm:ss</td>
<td>2008-11-04 06:46:20</td>
</tr>
<tr>
<td>Duration</td>
<td>A date that is equal to January 1st 1970 00:00:00 + the amount of time of the duration being stored</td>
<td><img src="image" alt="Duration Example" /></td>
</tr>
<tr>
<td>Choice</td>
<td>Returns the contents of the value field for the sys_choice record associated with that choice. See: Choice List for more information on returning the value associated with a particular item in a choice list.</td>
<td>&gt; &quot;2&quot; (Note that this value is is a string)</td>
</tr>
<tr>
<td>Journal</td>
<td>Returns a string of all entries made to that journal field. See Journal Fields for scripting of journal type fields</td>
<td>The web server is down &gt; The web server is down</td>
</tr>
</tbody>
</table>

Note: This date corresponds to the system time zone. If a different user time zone has been specified, the date and time value may appear different for that user.
Use business rules and client scripts to control field values

Implement both business rules and client scripts for a field to enable users to set record values properly using both forms and lists, and to see immediate changes to the values in forms as edits are made.

The problem with using only a client script or a business rule to control updates to a field is that fields can be changed on either a form or a list. Client scripts and UI policies run on forms only (client-side) and do not apply to list editing. Allowing list editing with client scripts running on fields in a form can result in incorrect data being saved to the record. For systems in which client scripts or UI policies apply to forms, either disable list editing or create appropriate business rules or access control to control the setting of values in the list editor. A side effect of this is that security measures implemented in client scripts are easy to circumvent. The user only needs to edit the field in a list.

Business rules on a form are not dynamic, the user must update the record for the change to be seen. This makes using client scripts the preferred method for controlling field values on forms.

When using both a business rule and client script to control field values, the update behavior is the same across the system. This means that updated values are not different depending on whether a list of form is used to make the change. This means that the same functionality must be implemented twice, once in a client script and once in a business rule or access control.

**Example 1**

An organization has a client script that sets the email address for a user to `first.last@company.com`. The administrator likes this because he can see the email address immediately when he enters the user's information. The administrator then performs a bulk import of users from a spreadsheet containing the users' first and last names. His expectation is that each user's email address will be set automatically, as they are when he edits the form. Since the client script runs only on the form (the interface to the record), it has no effect on data imported into the record from outside that interface, and no email addresses are created. To solve this problem, the administrator implements a business rule that runs when the import occurs and creates the email addresses.
Example 2

An organization wants to hide the **Priority** field on an incident form if the assignment group is Development. They create a UI policy on the incident form to do this, but their users can still see and edit the **Priority** field using the list editor. To rectify this, apply an access control to prevent read access to the **Priority** field when the assignment group is Development.

Using NULL as a field value

The string **NULL** has a particular role in scripts and is a reserved word.

A string containing the term **NULL** should not be used as a field value in import set **transform maps** or anywhere in the **First name** or **Last name** fields. The reserved word is **NULL** in all capital letters. A field with the value **Null** or **null**, for example, is acceptable. **NULL** should only be used to clear out a particular field.

Set a duration field value

Some examples of JavaScript that can be used to set the value of a duration field.

**Note:** Negative duration values are not supported.

The `dateDiff()` method

The **dateDiff(start, end, false)** method in **GlideSystem** enables you to set the duration value using a given start date/time and end date/time. An example to set the duration for the time a task was opened would be:

```
current.calendar_duration = gs.dateDiff(current.opened_at.getDisplayValue(), current.closed_at.getDisplayValue(), false);
```

The boolean value at the end of the call indicates the type of duration value to return:

- **false** = return the duration value
- **true** = return the difference between the two times in seconds; returned as a character or string value.

If you want to work with the value returned as a number to be used in date or duration aritmetic then use code similar to the following to convert the string value returned to a millisecond value:

```
var time = gs.dateDiff(<start_field>.getDisplayValue(), gs.nowDateTime(), true);
time = parseFloat(time) * 1000;
```

If you wanted to set a duration to the amount of time between some event and NOW, the code would be:

```
<duration_field> = gs.dateDiff(<start_field>.getDisplayValue(), gs.nowDateTime(), false);
```

The time values presented to `dateDiff()` are expected to be in the user's timezone and in the user's format. For any date or date/time field in the SNC database, use `getDisplayValue()` in GlideRecord to properly format the data.

If you want to do your own calculations and set the value of a duration field the best way would be to use `setDisplayValue()` to set the value for the field. For a duration field, `setDisplayValue()` expects a character string value with a format of **ddd hh:mm:ss** where **ddd** is the number of days for the duration.
For example, to set the duration field called \textit{u\_time\_to\_assign} in the current record to a value of 3 days 4 hours 30 minutes and 14 seconds you would:

\begin{verbatim}
current.u_time_to_assign.setDisplayValue('3 04:30:14');
\end{verbatim}

**Setting a default value of a duration field**

Setting the default value for a duration field is similar to the method used above.

**Setting the value of a duration field in a client script**

This script sets a duration field value in a client script. Replace \textit{u\_test\_duration} with the field name from your instance.

\begin{verbatim}
g_form.setValue('<duration_field>','11 01:02:03');
\end{verbatim}

**Calculating a duration and setting that value using a client script**

Here's an example of how to use the \texttt{gs.dateDiff} function to return a value and populate it using a client script. Create an \texttt{onChange} client script that includes the following code. You can modify this script if you need the calculation to happen in an \texttt{onLoad} script or some other way.

\begin{verbatim}
function onChange(control, oldValue, newValue, isLoading)
{var strt = g_form.getValue('<start_field>');var end =
 g_form.getValue('<end_field>');var ajax =new GlideAjax('AjaxDurCalc');
 ajax.addParam('sysparm_name','durCalc');
 ajax.addParam('sysparm_strt',strt);
 ajax.addParam('sysparm_end',end);
 ajax/XMLWait();var answer = ajax.getAnswer();
g_form.setValue('<duration_field>', answer);}
--Create a system script include file called \texttt{AjaxDurCalc} that handles the request. It may be reused for other functions as well.

var AjaxDurCalc =Class.create();
AjaxDurCalc.prototype= Object.extendsObject(AbstractAjaxProcessor,{
durCalc:function(){return
 gs.dateDiff(this.getParameter('sysparm_strt'),this.getParameter('sysparm_end'),false);}})
\end{verbatim}

**Changing the Duration Field Value**

If you manipulate a duration value with addition/subtraction of some amount of time then you can use the functions that allow you to get and set the numeric value of the duration. A unit of measure for a duration numeric value is milliseconds. Below is an example that adds 11 seconds the \textit{duration} field in the current record.

\begin{verbatim}
var timems = current.duration.dateNumericValue();
timems = timems + 11*1000;
current.duration.setDateNumericValue(timems);
\end{verbatim}
Format the Resolve Time

To format the Resolve Time or the Business Resolve Time fields as durations, which displays them as a duration instead of a large integer, add the following attribute to those fields:

```
format=glide_duration
```

Modify the dictionary entry for the field and add the attribute. If there is already an attribute there, multiple attributes must be separated by commas.

Set the Maximum Unit of Measurement

The `max_unit` dictionary attribute defines the maximum unit of time used in a duration. For example, if `max_unit=minutes`, a duration of 3 hours 5 minutes 15 seconds displays as 185 minutes 15 seconds. To set the maximum unit of duration measurement add the following dictionary attribute to the `duration` field:

```
max_unit=<unit>
```

Using regular expressions in scripts

Starting with Eureka Patch 5, an enhanced regex engine replaced the native JavaScript regex engine. JavaScript regular expressions automatically use the enhanced regex engine, which provides improved performance and supports all behaviors of standard regular expressions as defined by Mozilla JavaScript. The enhanced regex engine continues to support using Java syntax in regular expressions.

The `SNC.Regex` API is not available for scoped applications. When moving from Eureka Patch 4 and earlier versions, you should change your scripts to remove the `SNC.Regex` API and use standard JavaScript regular expressions.

For more information on JavaScript regular expressions, see the Mozilla JavaScript documentation on regular expressions and `RegExp`.

Using Java syntax in JavaScript regular expressions

The enhanced regex engine includes an additional flag to allow Java syntax to be used in JavaScript regular expressions.

Regular expressions with the additional flag work in all places that expect a regular expression, such as `String.prototype.split` and `String.prototype.replace`. To use Java syntax in a regular expression, use the Java inline flag `j`, for example `/(?ims)ex(am)ple/j`
### Table 1349: Extended regular expression flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>j</td>
<td>Defines a regular expression that executes using the Java regular expression engine. It can be used to access Java-only features of regular expressions (such as look behind, negative look behind) or to use Java regular expressions without translating them into JavaScript regular expressions. For example: [\text{var _regex = /ex(\text{am})ple/j;}]</td>
</tr>
</tbody>
</table>

#### Convert SNC Regex expressions to enhanced regex expressions

When you upgrade to Eureka Patch 5 or later releases, you should convert scripts that use the SNC.Regex API to use regular JavaScript expressions.

1. From the original expression, such as: \(\text{SNC.Regex("/expr/is")};\), create a new regular expression object using the pattern with the slashes stripped.

   ```javascript
   new RegExp('expr');
   ```

2. Move the `SNC.Regex` flags to the start of the expression using Java’s inline flag special construct.

   ```javascript
   new RegExp('(\text{is})expr');
   ```

3. Add the `j` flag to the `RegExp` to tell the engine to treat the expression as a Java expression.

   ```javascript
   new RegExp('(\text{is})expr', 'j');
   ```

   **Note:** If you know that the script being converted does not use Java syntax, it is not necessary to use the `j` flag.

4. Add the `g` flag to handle multiple matches or a global replace.

   ```javascript
   new RegExp('(\text{is})expr', 'jg');
   ```

#### Using SNC.Regex

```javascript
var r = new SNC.Regex('/world/');
var str = 'helloworld';
var replaced = r.replaceAll(str, 'there');
// replaced == 'hellothere'
```

#### Using a JavaScript regular expression

```javascript
var r = new RegExp('world', 'jg');
var str = 'helloworld';
var replaced = str.replace(r, 'there');
// replaced == 'hellothere'
```
Syntax editor

The syntax editor provides support for editing JavaScript scripts.

The syntax editor has these features.

- JavaScript syntax coloring, indentation, line numbers, and automatic creation of closing braces and quotes
- JavaScript support
- Script macros for common code shortcuts

This feature requires the Syntax editor plugin.

Figure 923: JavaScript editor

Context-sensitive API help

The syntax editor can display context-sensitive API information.

Context-sensitive help includes the ability to:

- List script elements that are valid at the cursor's location. The system displays suggestions in a pop-up window.
- Add a selected script element at the cursor's location. If the cursor is within or adjacent to a partial entry, the system completes the entry with the selected script element.
- View API documentation for a selected suggestion.
- View the expected parameters and format of the current script element.

If the cursor is adjacent to a text string, the system searches for script elements that start with this text string. For example, while the cursor is within or adjacent to the string GlideR, the system displays script elements such as:

- GlideRecord
- GlideRecordSecure
Context-sensitive suggestions are based on script type. For example, when working on a business rule, only suggestions from the server API and objects such as current and previous display. When working on a client script, the system only displays suggestions from the client API.

Syntax editor JavaScript support

The syntax editor provides editing functions to support editing JavaScript scripts.

**JavaScript editing functions**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Keyboard Shortcut</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Icon" /></td>
<td>N/A</td>
<td>Toggle Syntax Editor</td>
<td>Disables the syntax editor. Click the button again to enable the syntax editor.</td>
</tr>
<tr>
<td><img src="image2.png" alt="Icon" /></td>
<td>Access Key + R</td>
<td>Format Code</td>
<td>Applies the proper indentation to the script.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Icon" /></td>
<td>Access Key + C</td>
<td>Comment Selected Code</td>
<td>Comments out the selected code.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Icon" /></td>
<td>Access Key + U</td>
<td>Uncomment Selected Code</td>
<td>Removes comment codes from the selected code.</td>
</tr>
<tr>
<td><img src="image5.png" alt="Icon" /></td>
<td>N/A</td>
<td>Check Syntax</td>
<td>Checks the code for syntax errors. By default, the system automatically checks for syntax errors as you type in a script field. If an error or warning is found, the syntax editor displays a bullet beside the script line containing the error or warning. This check occurs on all script fields.</td>
</tr>
<tr>
<td><img src="image6.png" alt="Icon" /></td>
<td>Access Key + \</td>
<td>Start Searching</td>
<td>Highlights all occurrences of a search term in the script field and locates the first occurrence. Click the icon, then enter the search term and press Enter. You can use regular expressions enclosed in slashes to define the search term. For example, the term /a(3)/ locates aaa.</td>
</tr>
<tr>
<td><img src="image7.png" alt="Icon" /></td>
<td>Access Key + [</td>
<td>Find Next</td>
<td>Locates the next occurrence of the current</td>
</tr>
<tr>
<td>Icon</td>
<td>Keyboard Shortcut</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>search term in the script field. Use Start Searching to change the current search term.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access Key + ]</td>
<td>Find Previous</td>
<td>Locates the previous occurrence of the current search term in the script field. Use Start Searching to change the current search term.</td>
</tr>
<tr>
<td></td>
<td>Access Key + W</td>
<td>Replace</td>
<td>Replaces the next occurrence of a text string in the script field. 1. Click the icon, then enter the string to replace and press Enter. You can use regular expressions enclosed in slashes to define the string to replace. For example, the term /a(3)/ locates aaa. 2. Enter the replacement string and press Enter.</td>
</tr>
<tr>
<td></td>
<td>Access Key + ;</td>
<td>Replace All</td>
<td>Replaces all occurrences of a text string in the script field. 1. Click the icon, then enter the string to replace and press Enter. You can use regular expressions enclosed in slashes to define the string to replace. For example, the term /a(3)/ locates aaa. 2. Enter the replacement string and press Enter.</td>
</tr>
<tr>
<td></td>
<td>N/A</td>
<td>Save</td>
<td>Saves changes without leaving the current view. Use this button in full screen mode to save.</td>
</tr>
</tbody>
</table>
### JavaScript editing tips

- To fold a code block, click the minus sign beside the first line of the block. The minus sign only appears beside blocks that can be folded. To unfold the code block, click the plus sign.
- To insert a fixed space anywhere in your code, press Tab.
- To indent a single line of code, click in the leading white space of the line and then press Tab.
- To indent one or more lines of code, select the code and then press Tab. To decrease the indentation, press Shift + Tab.
- To remove one tab from the start of a line of code, click in the line and press Shift + Tab.

### JavaScript resources

Scripts use ECMA 262 standard JavaScript. Helpful resources include:

- Mozilla: [http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Reference](http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Reference)
- History and overview: [http://javascript.crockford.com/survey.html](http://javascript.crockford.com/survey.html)

### Syntax editor keyboard shortcuts and actions

The syntax editor offers keyboard shortcuts and actions to assist in writing code.

#### Table 1350: Syntax editor keyboard shortcuts and actions for writing code

<table>
<thead>
<tr>
<th>Keyboard shortcut or action</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyboard shortcut or action</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| Scripting assistance     | Displays a list of valid elements at the insertion point such as:  
| Control+Spacebar         | • Class names  
|                           | • Function names  
|                           | • Object names  
|                           | • Variable names  
|                           | Double-click an entry to add it to the script. | 

```javascript
(function execute){
var gr =
}
```
<table>
<thead>
<tr>
<th>Keyboard shortcut or action</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter a period character after a valid class name.</td>
<td>The system displays a list methods for the class. Double-click an entry to add it to the script.</td>
<td><img src="image" alt="Script" /></td>
</tr>
<tr>
<td>Enter an open parenthesis character after a valid class, function, or method name.</td>
<td>The system displays the expected parameters for the class or method. Enter the expected parameters as needed.</td>
<td><img src="image" alt="Script" /></td>
</tr>
<tr>
<td>Toggle full screen mode Control+M</td>
<td>Switches between displaying the form with the full screen and displaying it normally.</td>
<td><img src="image" alt="Script" /></td>
</tr>
<tr>
<td>Keyboard shortcut or action</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Format code</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <strong>Windows</strong>: Control+Shift+B</td>
<td>Formats the selected lines to improve readability.</td>
<td><img src="image" alt="Script example" /></td>
</tr>
<tr>
<td>• <strong>Mac</strong>: Command+Shift+B</td>
<td></td>
<td><img src="image" alt="Script example" /></td>
</tr>
<tr>
<td>Keyboard shortcut or action</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>Toggle comment</td>
<td>Adds or removes the comment characters // from the selected lines.</td>
<td><img src="image" alt="Script example" /></td>
</tr>
<tr>
<td>• <strong>Windows</strong>: Control+/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <strong>Mac</strong>: Command+/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyboard shortcut or action</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>Insert macro text</td>
<td>Inserts macro text at the current position.</td>
<td><img src="image" alt="Script Example" /></td>
</tr>
</tbody>
</table>

1. In the **Script** field, type the macro keyword text. For example `help`.
2. Press Tab.

**Search**
<table>
<thead>
<tr>
<th>Keyboard shortcut or action</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start search</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <strong>Windows</strong>: Control+F</td>
<td>Highlights all occurrences of a search term in the script field and locates the first occurrence. You can create <em>regular expressions</em> by enclosing the search terms between slash characters. For example, the search term <code>/a{3}/</code> locates the string <code>aaa</code>.</td>
<td><img src="image" alt="Script example" /></td>
</tr>
<tr>
<td>• <strong>Mac</strong>: Command+F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyboard shortcut or action</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Find next</strong></td>
<td></td>
<td><img src="image1" alt="Script" /></td>
</tr>
<tr>
<td>• Windows: Control+G</td>
<td>Locates the next occurrence of the current search term in the script field. Use Start Searching to change the current search term.</td>
<td></td>
</tr>
<tr>
<td>• Mac: Command+G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| <strong>Find previous</strong>           |             | <img src="image2" alt="Script" /> |
| • Windows: Control+Shift+G  | Locates the previous occurrence of the current search term in the script field. Use Start Searching to change the current search term. |         |
| • Mac: Command+Shift+G      |             |         |</p>
<table>
<thead>
<tr>
<th>Keyboard shortcut or action</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace</td>
<td>Replaces the next occurrence of a text string in the script field.</td>
<td><img src="image" alt="Script Example" /></td>
</tr>
</tbody>
</table>

- **Windows**: Control+E
- **Mac**: Command+E
<table>
<thead>
<tr>
<th>Keyboard shortcut or action</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace all</td>
<td>Replaces all occurrences of a text string in the script field.</td>
<td><img src="image" alt="Script" /></td>
</tr>
</tbody>
</table>

- **Windows**: Control+;  
- **Mac**: Command+;
<table>
<thead>
<tr>
<th>Keyboard shortcut or action</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td>Displays the list of Syntax Editor keyboard shortcuts.</td>
<td><img src="image1" alt="Editor Key Map" /></td>
</tr>
<tr>
<td>• Windows: Control+H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Mac: Command+H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show description</td>
<td>Display API documentation for the scripting element at the cursor's current location.</td>
<td><img src="image2" alt="Script" /></td>
</tr>
<tr>
<td>• Windows: Control+J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Mac: Command+J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyboard shortcut or action</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>Show macros</td>
<td>Displays the list of available Syntax Editor macros as text within the script field.</td>
<td><img src="image" alt="Script Editor" /></td>
</tr>
</tbody>
</table>

1. In the **Script** field, type `help`.
2. Press Tab.

**Navigate to a line number**

When the syntax editor is disabled, users can navigate to a specific line in the code using the Go to line icon (⏏).

1. Click the Go to line icon (⏏). A text field appears.
   
   **Note:** This icon is not available when the editor is enabled.

2. Enter a number in the field and then press **Enter**.

**Script syntax error checking**

All script fields provide controls for checking the syntax for errors and for locating the error easily when one occurs. The script editor places the cursor at the site of a syntax error and lets you search for errors in scripts by line number.
The script editor notifies you of syntax errors in your scripts in the following situations.

- Save a new record or update an existing record. A banner appears at the bottom of the editor showing the location of the first error (line number and column number), and the cursor appears at the site of the error. Warnings presented at `Save` or `Update` show only one error at a time.

- Click the syntax checking icon before saving or updating a record. A banner appears at the bottom of the editor showing the location of all errors in the script, and the cursor appears at the site of the first error.
Figure 926: Script syntax error

Searching for errors by line

To locate the exact position of the error in a large script, click the Go to line icon.

This feature is particularly useful when you are encounter a syntax error in a log file rather than in the ServiceNow record itself. In this case, you can navigate to the record and search for errors by line number. In the dialog box that appears, enter the line number of an error, and then click OK. Your view moves to the site of the error, and the cursor marks the correct line and column.

**Note:** For this feature to function, you must disable the Syntax Editor.

Figure 927: Go to script error
Syntax editor macros

Script macros provide shortcuts for typing commonly used code. To insert macro text into a script field, enter the macro keyword followed by the Tab.

vargr

- **Description**: Inserts a standard GlideRecord query for a single value.
- **Output**:

```javascript
var gr = new GlideRecord('');
gr.addQuery('name', 'value');
gr.query();
if (gr.next()) {
    
}
```

vargror

- **Description**: Inserts a GlideRecord query for two values with an OR condition.
- **Output**:

```javascript
var gr = new GlideRecord('');
var qc = gr.addQuery('field', 'value1');
qc.addOrCondition('field', 'value2');
gr.query();
while (gr.next()) {
    
}
```

for

- **Description**: Inserts a standard recursive loop with an array.
- **Output**:

```javascript
for (var i=0; i< myArray.length; i++) {
    //myArray[i];
}
```

info

- **Description**: Inserts a GlideSystem information message.
• **Output:**

```javascript
gs.addInfoMessage("";
```

**method**

• **Description:** Inserts a blank JavaScript function template.

• **Output:**

```javascript
/

* Description:

* Parameters:

* Returns:

: function() {


},
```

**doc**

• **Description:** Inserts a comment block for describing a function or parameters.

• **Output:**

```javascript
/**

* Description:

* Parameters:

* Returns:

*/
```

---

**Script macro maintenance**

Administrators can define new script macros or modify existing script macros.

Role required: admin

Script macros provide shortcuts for typing commonly used code. Several script macros are available by default. Administrators can define new or modify existing script macros.

1. Navigate to **System Definition > Syntax Editor Macros.**
2. Click **New** or select the macro to edit.
3. Define the macro details with the fields listed in the table below.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Macro keyword text users type to insert macro text.</td>
</tr>
<tr>
<td>Comments</td>
<td>Description of the macro. This text appears when the user types help.</td>
</tr>
</tbody>
</table>
Syntax editor plugin

Syntax editor plugin enables use of the syntax editor.

The syntax editor enables the following features for all script fields:

- JavaScript syntax coloring, indentation, line numbers, and automatic creation of closing braces and quotes
- Code editing functions
- Code syntax checking
- Script macros for common code shortcuts

![JavaScript syntax editor](image)

Figure 928: JavaScript syntax editor

The syntax editor can be disable/enable by modifying the glide.ui.javascript_editor property in the sys_properties.list. In addition, administrators can configure the syntax editor to show error and warning indicators next to a line of code that contains an error by modifying the glide.ui.syntax_editor.show_warnings_errors property. For information on the sys_properties.list, refer to Available system properties.

**Note:** Administrators can disable or enable the syntax editor for all users, regardless of user preference.

Session debug

Enable session debugging to display debugging messages in the user interface.
After being enabled, session debugging is active for the duration of the user session or until disabled. The system provides the following session debugging options.

**Table 1352: Session debug options**

<table>
<thead>
<tr>
<th>Debug option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable All</td>
<td>Displays all available debugging messages.</td>
</tr>
<tr>
<td>Disabled All</td>
<td>Stops displaying all debugging messages.</td>
</tr>
<tr>
<td>Debug Business Rule</td>
<td>Display debugging messages for business rules. If there are business rules from multiple applications affecting a table or record, the system displays which application the business rule comes from.</td>
</tr>
<tr>
<td>Debug Business Rule (Details)</td>
<td>Displays debugging messages for business rules and any changes made by business rules. If there are business rules from multiple applications affecting a table or record, the system displays which application the business rule comes from.</td>
</tr>
<tr>
<td>Debug Log</td>
<td>Displays all log entries.</td>
</tr>
<tr>
<td>Debug SQL</td>
<td>Displays debugging messages for SQL queries.</td>
</tr>
<tr>
<td>Debug SQL (Detailed)</td>
<td>Displays debugging messages for SQL statements and any changes made by SQL statements.</td>
</tr>
<tr>
<td>Debug Security</td>
<td>Displays debugging messages for access controls. If there are access controls from multiple applications affecting a table or record, the system displays which application the access controls come from.</td>
</tr>
<tr>
<td>Debug Escalations</td>
<td>Displays debugging messages for SLA and SLO escalations.</td>
</tr>
<tr>
<td>Debug Text Search</td>
<td>Displays debugging messages for search result relevance and indexing.</td>
</tr>
<tr>
<td>Debug UI Policies</td>
<td>Displays debugging messages for UI policies.</td>
</tr>
<tr>
<td>Disable UI Policies Debug</td>
<td>Stops displaying debugging messages for UI policies.</td>
</tr>
<tr>
<td>Debug Data Policies</td>
<td>Displays debugging messages for data policies.</td>
</tr>
<tr>
<td>Debug Quotas</td>
<td>Displays debugging messages for transaction quotas.</td>
</tr>
<tr>
<td>Debug Homepage Render</td>
<td>Displays debugging messages for homepages.</td>
</tr>
<tr>
<td>Debug Scopes</td>
<td>Displays debugging messages for entering or exiting application scopes when running script objects.</td>
</tr>
</tbody>
</table>
### Debugging applications

Application developers can display debug messages about configuration records to help them troubleshoot issues.

The system offers the following debugging options to help application developers determine how applications affect configuration records.

<table>
<thead>
<tr>
<th>Debugging option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug Business Rule</td>
<td>Use this module to determine which application's business rules are running against tables. The system only displays application information if business rules from different application scopes run on the same table.</td>
</tr>
<tr>
<td>Debug Business Rule (Details)</td>
<td>Use this module to determine the results of running business rules against tables. The system only displays application information if business rules from different application scopes run on the same table.</td>
</tr>
<tr>
<td>Debug Security</td>
<td>Use this module to determine which application's access controls apply to a given table or record.</td>
</tr>
<tr>
<td>Debug Scopes</td>
<td>Use this module to determine the application scope context in which a script runs. Since one script can call another script it is possible to have multiple application scope context changes while running a series of scripts.</td>
</tr>
<tr>
<td>Enable Session Debug</td>
<td>Use this related link to enable the generation of log messages for a particular application. Application scripts that use GlideSystem logging methods will generate output to the log at the indicated verbosity level.</td>
</tr>
</tbody>
</table>

When multiple applications contribute to the debug output, the system adds a new section called **Apps** to the display a list of the applications writing to the session log. Clicking on the check box next to the application name hides or displays the application's associated debug messages.
Figure 929: Sample application debug output of business rules

Debugging scopes

Application developers can use the **Debug Scopes** module to display information about when the system switches between custom applications to run server-side scripts.

When enabled, the system displays a message whenever the system switches to a custom application to run a server-side script.
Geneva    ServiceNow    ServiceNow Platform

Figure 930: Sample debug scopes output from the incident table

Every time the system runs a server-side script object it enters the script's scope context. When the script finishes running, the script exits the scope context. The debugging messages track changes to the script scope context.

The debugging message displays a greater than character > each time the system enters a script object's context, and displays a less than character < every time the system exits a script object's context. In cases where one script calls another the debugging message adds another greater than character to the path for each call. For example, if a business rule calls a script include, which in turn calls another script object there would three characters in the path such as:

```
> Entering scope [x_app_one]
>> Entering scope [x_app_two]
>>> Entering scope [x_app_three]
```

**Note:** The system does not display entering or exiting messages for script objects in the global scope.

Application developers may want to enable other debugging options to in conjunction with this option to see information about the possible source of the server-side script such as Debug Business Rule.

**Debugging Business Rules**

Debugging business rules can be achieved with resources available in the ServiceNow product.
1. Tools
The first step in the process is to identify tools which will help you figure out what’s wrong.

<table>
<thead>
<tr>
<th>Debugging Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Dictionary</td>
<td>Navigate to System Definition &gt; Dictionary. The dictionary provides a list of</td>
</tr>
<tr>
<td></td>
<td>all tables within your instance and can be invaluable when trying to locate</td>
</tr>
<tr>
<td></td>
<td>information.</td>
</tr>
<tr>
<td>System Log</td>
<td>Navigate to System Logs &gt; System Log. You can place alert statements in your</td>
</tr>
<tr>
<td></td>
<td>business rule which can write information to the log.</td>
</tr>
<tr>
<td>Debug Business Rule (Details)</td>
<td>Navigate to System Diagnostics &gt; Debug Business Rule (Details). This debugg-</td>
</tr>
<tr>
<td></td>
<td>ing module displays the results business rules. Use this module to see if</td>
</tr>
<tr>
<td></td>
<td>conditions are being met and values are being set as expected.</td>
</tr>
<tr>
<td>Alert Messages</td>
<td>There are several system functions which will allow you to print messages to</td>
</tr>
<tr>
<td></td>
<td>the page, the field or the log file. See Scripting Alert, Info, and Error</td>
</tr>
<tr>
<td></td>
<td>Messages.</td>
</tr>
<tr>
<td>Business Rule Examples</td>
<td>Sometimes you can find what you’re looking for in scripts others have written,</td>
</tr>
<tr>
<td></td>
<td>including business rule error messages, or by building an OR query.</td>
</tr>
<tr>
<td>GlideRecord Information</td>
<td>This is the basic syntax used to query the database for information. See</td>
</tr>
<tr>
<td></td>
<td>Using GlideRecord to Query Tables. GlideRecord also includes Aggregation</td>
</tr>
<tr>
<td></td>
<td>Support.</td>
</tr>
</tbody>
</table>

2. Variables
The next step is to gain some insight into the behavior of your business rule. For every action except an insert you will more than likely use a query to get your record(s).

```javascript
var rec = new GlideRecord('incident');
rec.addQuery('active',true);
rec.query();
while (rec.next()) {
    gs.print(rec.number + ' exists');
}
```

To verify whether your query is actually returning records you can use `gs.addInfoMessage` to display information at the top of the screen.

```javascript
var rec = new GlideRecord('incident');
rec.addQuery('active',true);
rec.query();
gs.addInfoMessage("This is rec.next: " + rec.next());
while (rec.next()) {
    gs.print(rec.number + ' exists');
}
```
If your query returns no records you will see this:

```
This is rec.next: false
```

Use this technique to verify every variable within your business rule contains expected values.

Tip: If necessary, break your script down into individual pieces and verify each piece works separate from the whole and then put them all back together one step at a time.

3. Locating Information

The last step is to make sure you know where to find the information your rule is looking for.

In the ServiceNow application, one table can extend another table. This means when searching for information, you might need to query the parent table for the extended table’s sys_id to find what you seek.

Example A good example is the sc_task table, which extends the task table. The script below queries the extended table (sc_task) for the current sys_id and then query the parent table (task) for records with the matching sys_id, and then prints out the work notes field.

```javascript
var kids = new GlideRecord('sc_task');
kids.query();
gs.addInfoMessage("This is requested item number: " + current.number);
gs.print("This is the requested item number: " + current.number);

while (kids.next()) {
    var parents = new GlideRecord('task');
    parents.addQuery('sys_id', '=?', kids.sys_id);
    parents.query();

    while(parents.next()) {
        gs.addInfoMessage("This is task number: " + parents.number);
        gs.print("This is task number: " + parents.number);
        gs.addInfoMessage("These are the work notes: " + parents.work_notes);
        gs.print("These are the work notes: " + parents.work_notes);
    }
}
```

### Debugging classifications

You must add a system property to enable classification debugging.

#### Debugging classifications

The resulting log entries list the name of each classifier that runs, along with all the names and values that are available to the criteria in the classifier. To log debugging information about classifications, add the following system property:
<table>
<thead>
<tr>
<th>System Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glide.discovery.debug.classification</td>
<td>Enables debugging information for process classification.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Type:</strong> true</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default Value:</strong> false</td>
</tr>
<tr>
<td></td>
<td>• <strong>Location:</strong> Add to the System Properties [sys_properties] table</td>
</tr>
</tbody>
</table>

Field Watcher

The Field Watcher tool tracks and displays all actions that the system performs on a selected form field. Administrators can use the field watcher to figure out what happens to the field and how the value of the field changes when an event such as the firing of a business rule or enforcement of a data policy, takes place. Administrators can also impersonate non-admin users to debug what happens when those users make changes on an instance. Only one field can be watched at a time. Non-admin users with the impersonator role have access to the field watcher feature.

How the Field Watcher Works

The Field Watcher tool logs activity when any of the following events occur on a field:

• The default value is set on the field.
• User access rights for the field change due to an ACL or dictionary setting.
• A data policy prevents the value from being set.
• A reference qualifier query of the field value executes.
• A UI policy changes a field to or from read-only, visible, mandatory, or editable.
• A dependent value in another field restricts field choices.
• The value of the field is set or changed based on:
  • Assignment rules
  • Actions from an engine, such as the workflow engine
  • Business rules
  • User entries
  • Client scripts
  • UI actions

**Note:** The field watcher works only on form fields. It cannot be used on list fields. Also, field watcher is not available on password-protected fields or encrypted fields.

Use Field Watcher

1. Navigate to the form for which you want to view field-level debugging information.
2. Activate Field Watcher by right-clicking any field label on a form and select **Watch - '<field name>'**.

The debug icon ( ![Debug Icon](debug_icon.png) ) appears next to the field label. From this point on, the Field Watcher records every action taken on the selected field. For example, if you are watching a Priority field, if the priority...
is changed from Moderate to Low and the record is updated, the Field Watcher will display information about that change.

3. View the Field Watcher log by clicking the debug icon. A new pane opens at the bottom of the screen, showing a Field Watcher tab. It may also show tabs for JavaScript Logging and JavaScript Debugger.

4. Click the Field Watcher tab, if needed.

5. Stop watching a field by right-clicking the field and selecting Unwatch - <field name>. To watch another field, right-click that field and select Watch - <field name>.

6. Clear the Field Watcher log by clicking the clear log button.

7. Resize the Field Watcher pane by dragging the splitter bar up or down. Dragging the splitter bar to the bottom of the screen closes the Field Watcher pane. Reopen the pane by clicking the debug icon again.

Field Watcher tab details

The Field Watcher displays field information and configuration options. The left-side of the Field Watcher tab shows the following information for the watched field.

- **Table**: table to which the field belongs.
- **Element**: field label.
- **Type**: type of data stored in the field.
- **Dependent**: field on which the current field depends.
- **Reference**: table from which the field’s value originates, if applicable.
- **Reference Qual**: reference qualifiers that may be restricting data on the field.
- **Attributes**: attributes on the field as specified in the dictionary entry for that field.

On the right-side of the Field Watcher tab, select the types of activity information you want to see for the selected field. Clear the check box for any type of information that is not needed.

**Viewing information for the watched field**

When information for a watched field is changed and the record is updated, the Field Watcher tab displays relevant information at the bottom.

### Figure 931: Field watcher viewing data

Field watcher information includes:

- **Timestamp**: time the field was changed using the HH:MM:SS (ms) format.
  - Orange text: server-side changes, such as ACLs.
  - Blue text: client-side changes, such as client scripts.

- **Type of object that changed the field and its associated name**: The type of item that changed on the field; for example, CLIENT SCRIPT, BUSINESS RULE, or ACL. In the case of scripts, business rules, or other configuration-type fields, field watcher displays the name of the script or business rule that changed the field, if any. Click the name to go directly to the record for that item.

- **Old and new values**: The old and new values for the field, if the value changed. Field watcher does not record the value if it was inserted in the form by default at the time the record was created.

- **Additional information**: Call tracing information, such as the name of the script engine or workflow that changed the field. Click the plus icon to expand the selection.
Watching a hidden field

Administrators may need to watch a hidden field.

1. Use the dictionary to determine the column name of the field.
2. Elevate privileges to the security_admin role.
3. Navigate to **System Definition > Scripts Background**.
4. In Run script (JavaScript executed on server), enter the following command:

   ```javascript
   gs.getSession().setWatchField("hidden_field");
   ```

   Replace `hidden_field` with the column name of the hidden field.
5. Navigate to the form containing the missing field.
   
   The Field Watcher tab output contains information about the hidden field.

Example: Watching the incident priority

The following example shows what happens to the Priority field on the incident form when both the Impact and Urgency fields change.

The Incident form has two client-side data lookups change the priority. Additionally, server-side ACLs and the data lookup engine fire when the record is saved. Finally, a client-side UI policy sets the Priority field back to read-only, which is the default setting.

**Table 1355: Watching the Incident Priority**

<table>
<thead>
<tr>
<th>Original Values</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority: 1 - Critical</td>
<td></td>
</tr>
<tr>
<td>Impact: 1 - High</td>
<td></td>
</tr>
<tr>
<td>Urgency: 1 - High</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Change</th>
<th></th>
</tr>
</thead>
</table>

1. The user changes the Impact value to 3 - Low.
2. The priority automatically changes to 3 - Moderate based on the Priority Lookup data lookup definition used by default in ServiceNow incidents.

**Note**: At this point, the record has not been saved.

<table>
<thead>
<tr>
<th>Second Change</th>
<th></th>
</tr>
</thead>
</table>
**Original Values**

1. The user changes the Urgency value to 2 - Medium.
2. The priority automatically changes to 4 - Low based on the same Priority Lookup data lookup definition.
3. The user saves the record by right-clicking the form header and choosing Save.

---

**Figure 932: Field watcher example**

**Note:** The values that change from 1 to 3, and then from 3 to 4, refer to the numerical values in the choice list.

**Tutorials**

The following video demonstrates how to perform field-level debugging with the field watcher.
**JavaScript log**

JavaScript that runs on the browser, such as client scripts, can include a call to jslog() to send information to the JavaScript Log. Users with the admin role can access this log.

1. Open the JavaScript log by navigating to the appropriate location for your version of the UI.

<table>
<thead>
<tr>
<th>UI16</th>
<th>1. Click the gear icon in the banner frame.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. Click the Developer section.</td>
</tr>
<tr>
<td></td>
<td>3. Toggle the JavaScript Log and Field Watcher switch.</td>
</tr>
<tr>
<td>UI15</td>
<td>1. Click the gear icon in the banner frame.</td>
</tr>
<tr>
<td></td>
<td>2. Click JavaScript Log and Field Watcher.</td>
</tr>
<tr>
<td>UI11</td>
<td>Click the debug icon ( <img src="image" alt="debug icon" />) in the banner frame.</td>
</tr>
</tbody>
</table>

A new pane opens at the bottom of the screen. It shows the JavaScript Log tab and may also show the Field watcher tab.

2. If needed, select the JavaScript Log tab.

3. Click the clear icon ( ![clear icon](image)) to clear the contents of the log, as needed.

**Debug UI policies**

Enabling the glide.ui.ui_policy_debug property lets you monitor the processing of UI actions.

Here are some sample log events from an incident policy that sets fields to read-only if the incident_state is closed.

```plaintext
GlideFieldPolicy: Evaluating condition
GlideFieldPolicy:       incident_state (7) = 7 -> true
GlideFieldPolicy: --->>> TRUE
GlideFieldPolicy: Setting opened_at disabled to true
GlideFieldPolicy: Setting opened_by disabled to true
GlideFieldPolicy: Setting closed_at disabled to true
GlideFieldPolicy: Setting closed_by disabled to true
GlideFieldPolicy: Setting company disabled to true
```

**Writing to the debug log**

To write to the debug log in your client-side JavaScript, or UI policies, make a call to the global function jslog().

An example of using jslog() in JavaScript:

```javascript
function logData (r ) {
    lastLogDate  = r. responseXML. documentElement. getAttribute
    ( "last_log_entry" ) ; var items  = r. responseXML. getElementsByTagName
    "log" ) ;
    jslog ( "response=" + r. responseText ) ; }
```
Additionally, when client scripts run, the name of the client script and timing information is displayed. This can be useful in determining which scripts are running and whether they are impacting performance.

**JavaScript debug window**

The JavaScript debug window appears in a bottom pane of the user interface when an administrator turns on debugging.

Use the debug window to access these tools.

- **JavaScript Log**: JavaScript that runs on the browser, such as client scripts, can include a call to `jslog()` to send information to the JavaScript log.
- **Field Watcher**: a tool that tracks and displays all actions that the system performs on a selected form field.

![JavaScript Log and Field Watcher](image)

**Figure 933: JavaScript debug window**

**Using the JavaScript debug window**

The JavaScript debug window enables access to the JavaScript Log and the Field Watcher tools.

The steps to access the JavaScript debug window depend on which UI version you are using.

1. Open the JavaScript debug window by navigating to the appropriate location for your version of the UI.

| UI16          | 1. Click the gear icon in the banner frame.  
|              | 2. Click the Developer section.  
|              | 3. Toggle the JavaScript Log and Field Watcher switch. |

| UI15          | 1. Click the gear icon in the banner frame.  
|              | 2. Click JavaScript Log and Field Watcher. |
Click the debug icon ( ✗ ) in the banner frame.

The JavaScript debug window opens at the bottom of the screen. The tab that is currently active in the window is the last tab that was active when the window was closed.

2. Click a tab to use one of the debug window features.
   - JavaScript Log
   - Field Watcher

Server scripts

Server scripts run on the server or database. They can change the appearance or behavior of ServiceNow or run as business rules when records and tables are accessed or modified.

Server-side Glide APIs (Application Programming Interfaces) provide classes and methods that you can use in scripts to perform server-side tasks.

Immediately invoked function expressions

An immediately invoked function expression (IIFE) is both declared and invoked within the same script field.

The system uses immediately invoked function expressions when a script runs in a single context, such as in a transform map script. Functions that run from multiple contexts use script includes instead.

By enclosing a script in an immediately invoked function expression you can:
   • Ensure the script does not impact other areas of the product, such as by overwriting global variables.
   • Pass useful variables or objects as parameters.
   • Identify function names in stack traces.
   • Eliminate having to make separate function calls.

An immediately invoked function expression follows this format:

```javascript
(function functionName(parameter){
    //The script you want to run
})(value);//Note the parenthesis indicating this function should run.
```

You can declare functions within the immediately invoked function expression. These inner functions are accessible only from within the immediately invoked function expression.

```javascript
(function functionName(parameter){
    function helperFunction(parameter){//return some value
        var value = helperFunction(parameter);//Valid function call.
        //perform any other script actions
    })('value');

    var value2 = helperFunction(parameter);//Invalid. This function is not accessible from outside the self-executing function.
}
```
Business rules

A business rule is a server-side script that runs when a record is displayed, inserted, updated, or deleted, or when a table is queried.

Use business rules to accomplish tasks like automatically changing values in form fields when certain conditions are met, or to create events for email notifications and script actions.

Note: Business rules can make use of scripts to take actions on records in the database. However, there are several other scripting options available on the platform, such as client scripts and UI actions.

How business rules work

To configure business rules, you first need to determine when the business rule should run and what action it should take.

When business rules run

Business rules run based on two sets of criteria:

- The time that the business rule is configured to run relative to a record being modified or accessed.
- The database operation that the system takes on the record.

The following options are provided to determine the time the business rule should run:

Table 1356: Time the business rule should run

<table>
<thead>
<tr>
<th>Option</th>
<th>When the rule runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>After the user submits the form but before any action is taken on the record in the database.</td>
</tr>
<tr>
<td>After</td>
<td>After the user submits the form and after any action is taken on the record in the database.</td>
</tr>
<tr>
<td>Async</td>
<td>When the scheduler runs the scheduled job created from the business rule. The system creates a scheduled job from the business rule after the user submits the form and after any action is taken on the record in the database.</td>
</tr>
<tr>
<td>Display</td>
<td>Before the form is presented to the user, just after the data is read from the database.</td>
</tr>
</tbody>
</table>

Note: Asynchronous business rules do not have access to the previous version of a record. Therefore, the changes(), changesTo(), and changesFrom() GlideElement methods do not work with async rules.

The following options are provided to determine the database operation that the system takes on the record:
Table 1357: Database operation that the system takes on the record

<table>
<thead>
<tr>
<th>Option</th>
<th>When the rule runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>When the user creates a new record and the system inserts it into the database.</td>
</tr>
<tr>
<td>Update</td>
<td>When the user modifies an existing record.</td>
</tr>
<tr>
<td>Query</td>
<td>Before a query for a record or list of records is sent to the database. Typically you should use <code>query</code> for before business rules. See <a href="#">Before-Query</a> example.</td>
</tr>
<tr>
<td>Delete</td>
<td>When the user deletes a record.</td>
</tr>
</tbody>
</table>

This image shows when different types of business rules run:

![Business rule processing flow](image)

**Figure 934: Business rule processing flow**

**Note:** Business rules apply consistently to records regardless of whether they are accessed through forms, lists, or web services. This is one major difference between business rules and client scripts, which apply only when the form is edited.
Business rule actions

Business rules can perform a variety of actions. Common types of actions are:

- Changing field values on a form that the user is updating. Field values can be set to specific values available for that field, values copied from other fields, and relative values determined by the user's role.
- Displaying information messages to the user.
- Changing values of child tasks based on changes to parent tasks.
- Preventing users from accessing or modifying certain fields on a form.
- Aborting the current database transaction. For example, if certain conditions are met, prevent the user from saving the record in the database.

Administrators can set field values, create information messages, and abort transactions without writing a script.

Business rules in scoped applications

Every business rule is assigned to either a private application scope or to the global scope.

The types of business rules you can create and how you can access those rules varies depending on the scope of the business rule and the scope of the table it runs on.

Note: The term **global** can refer to two different aspects of a business rule: the table it runs on and the scope it runs in. Business rules can either run on specific tables or be global. In addition, they can be in the global scope or in a private application scope.

Business rules on specific tables

Most business rules run on a specific table, which is defined in the **Table** field. You can create business rules on tables in the same scope and on tables that allow configuration records from another application scope.

For tables that are in a different scope than the business rule record, the types of rules are limited.

- You can create a rule where **When is async** with any of the following options:
  - **Insert**, **Update**, and **Delete** database operations. You cannot select **Query**.
  - **Set field values** actions and scripts (the **Script** field).
- You can create a rule where **When is before** with any of the following options:
  - **Insert**, **Update**, and **Delete** database operations. You cannot select **Query**.
  - **Set field values** actions only. You cannot write scripts and you cannot abort the database transaction.
- You cannot create any other types of business rules on tables in a different scope.

Business rules on specific tables cannot be accessed by other business rules or scripts.

Global business rules

Warning: Consider using script includes instead of global business rules. Script includes load only on request while global business rules load on every page in the system.
Global business rules are business rules where the **Table** field is set to **Global**. Global business rules may be accessible on multiple tables and from other scripts, depending on their scope protection. For a global business rule, define the scope protection by setting the **Accessible from** field:

- **This application scope only**: prevents applications in a different scope than the business rule from calling this business rule.
- **All application scopes**: allows any application to call this business rule.

**Note**: Global business rules do not support domain separation.

### Scripts in scoped business rules

When you write a script in a business rule, you can access:

- Any script includes and global business rules in the same scope as the business rule.
- Script includes and global business rules that allow applications in a different scope to call them. To call functions from another scope, you must specify the scope of the function.
- For business rules in the global scope, you can access the system APIs (**GlideRecord**, **GlideSystem**, etc.) For more information on global APIs see the [legacy APIs on the developer portal](#).
- For business rules in a unique scope, you can access the scoped system APIs only (**ScopedGlideRecord**, **ScopedGlideSystem**, etc.) For more information on scoped APIs see the [scoped APIs on the developer portal](#).

For more information on scoped applications, see [scoped application scripting on the developer portal](#).

### Create a business rule

You can create any type of business rule from the Business Rule form.

1. Navigate to **System Definition > Business Rules**.
2. Click **New**.
3. Fill in the fields, as appropriate.

**Note**: You might need to configure the form to see all fields.

#### Table 1358: Business Rule Form Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the business rule.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the table that the business rule runs on.</td>
</tr>
</tbody>
</table>

**Note**: The list shows only tables and database views that meet the scope protections for business rules starting with the Fuji release. Business rules defined for a database view can run only on **Query**. A business rule for a database view cannot run on insert, update, or delete.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application that contains this business rule.</td>
</tr>
<tr>
<td>Accessible from</td>
<td>Scope protection for a global business rule.</td>
</tr>
<tr>
<td>Note:</td>
<td>This field is visible only when the Table field is set to Global. It does not apply to rules that run on specific tables.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to enable the business rule.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Select this check box to see the advanced version of the form.</td>
</tr>
<tr>
<td>When to run</td>
<td></td>
</tr>
<tr>
<td>When</td>
<td>[Advanced] Select when this business rule should execute: display, before, async, or after the database operation is complete.</td>
</tr>
<tr>
<td>Note:</td>
<td>Consider setting the Priority for async business rules as the system uses this value when creating the associated scheduled job.</td>
</tr>
<tr>
<td>Order</td>
<td>[Advanced] Enter a number indicating the sequence in which this business rule should run. If there are multiple rules on a particular activity, the rules run in the order specified here, from lowest to highest.</td>
</tr>
<tr>
<td>Insert</td>
<td>Select this check box to execute the business rule when a record is inserted into the database.</td>
</tr>
<tr>
<td>Update</td>
<td>Select this check box to execute the business rule when a record is update.</td>
</tr>
<tr>
<td>Delete</td>
<td>[Advanced] Select this check box to execute the business rule when a record is deleted from the database.</td>
</tr>
<tr>
<td>Query</td>
<td>[Advanced] Select this check box to execute the business rule when a table is queried.</td>
</tr>
<tr>
<td>Filter Conditions</td>
<td>Use the condition builder to determine when the business rule should run based on the field values in the selected Table. You can also use the Condition field to build a condition with a script.</td>
</tr>
<tr>
<td>Role Conditions</td>
<td>Select the roles that users who are modifying records in the table must have for this business rule to run.</td>
</tr>
<tr>
<td>Actions</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Set field values | Set values for fields in the selected Table using the choice lists:  
  • The field  
  • The assignment operator:  
    • To: An exact value  
    • Same as: The value of another field  
    • To (dynamic): A value relative to the user configuring the business rule or a user with a specific role  
  • The value                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Add message   | Select this check box and enter a message that appears when this business rule is run                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Abort action  | Select this check box to abort the current database transaction. For example, on a before insert business rule, if the conditions are met, do not insert the record into the database.  
  If you select this option, you cannot perform additional actions on the record, such as setting field values and running scripts.  
  You can still display a message to users by selecting the Add message check box and composing the message.                                                                                                                                                                                                                                                                                                                                                   |
| Advanced      | [Advanced] Create a JavaScript conditional statement to specify when the business rule should run. By adding the condition statement to this field, you tell the system to evaluate the condition separately and run the business rule only if the condition is true.  
  If you decide to include the condition statement in the Script field or if you use the condition builder, leave this field blank.  
  To have the instance reevaluate the condition statement a second time before running an async business rule, add the system property glide.businessrule.async_condition_check and set the value to true.                                                                                                                                                                                                                          |
### Field Description

**Script**

[Advanced] Create a script that runs when the defined condition is true. The system automatically populates this field with a function name that matches the **When** value.

- onAfter
- onAsync
- onBefore
- onDisplay

**Note:** The function name must match the **When** value.

For more information and examples, see Scripting in Business Rules.

<table>
<thead>
<tr>
<th>Related list: Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Versions</td>
</tr>
</tbody>
</table>

Shows all versions of the business rule. Use this list to compare versions or to revert to a previous version.

4. Click **Submit**.

## Global variables in business rules

Predefined global variables are available for use in business rules. Use the following predefined global variables to reference the system in a business rule script.

<table>
<thead>
<tr>
<th>Global variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>current</td>
<td>The current state of the record being referenced. <em>Check for null</em> before using this variable.</td>
</tr>
<tr>
<td>previous</td>
<td>The state of the referenced record prior to any updates made during the execution context, where the execution context begins with the first update or delete operation and ends after the script and any referenced business rules are executed. If multiple updates are made to the record within one execution context, <strong>previous</strong> will continue to hold the state of the record before the first update or delete operation. Available on update and delete operations only. Not available on async operations. <em>Check for null</em> before using this variable.</td>
</tr>
<tr>
<td>g_scratchpad</td>
<td>The scratchpad object is available on display rules, and is used to pass information to the client to be accessed from client scripts.</td>
</tr>
<tr>
<td>system (or gs)</td>
<td>References to <em>GlideSystem</em> functions.</td>
</tr>
</tbody>
</table>
The variables current, previous, and g_scratchpad are global across all business rules that run for a transaction.

**Prevent null pointer exceptions**

In some cases, there may not be a current or previous state for the record when a business rule runs, which means that the variables will be null. To check for null before using a variable, add the following code to your business rule:

```java
if (current == null) // to prevent null pointer exceptions.
    return;
```

**Define variables**

User-defined variables are globally scoped by default. If a new variable is declared in an order 100 business rule, the business rule that runs next at order 200 also has access to the variable. This may introduce unexpected behavior.

To prevent such unexpected behavior, always wrap your code in a function. This protects your variables from conflicting with system variables or global variables in other business rules that are not wrapped in a function. Additionally, variables such as current must be available when a function is invoked in order to be used.

The following script is vulnerable to conflicts with other code. If the variable gr is used in other rules, the value of the variable may unexpectedly change.

```java
var gr = new GlideRecord('incident');
gr.query();
while(gr.next()) {
   //do something
}
```

When this script is wrapped in a function, the variable is available only within the function and does not conflict with other functions using a variable named gr.

```java
myFunction();
function myFunction() {
   var gr = new GlideRecord('incident');
gr.query();
while(gr.next()) {
   //do something
}

```

**Compare date fields in a business rule**

It is possible to compare two date fields or two date and time fields in a business rule.

It is possible to compare two date fields or two date and time fields in a business rule, and abort a record insert or update if they are not correct. For example, you may want a start date to be before an end date.

The following is an example script:

```java
if ((!current.u_date1.nil()) && (!current.u_date2.nil())) {
   var start = current.u_date1.getGlideObject().getNumericValue();
   var end = current.u_date2.getGlideObject().getNumericValue();
```
if (start > end) {
    gs.addInfoMessage('start must be before end');
    current.u_date1.setError('start must be before end');
    current.setAbortAction(true);
}

As a good practice, make the business rule a before rule for insert and update actions. In the example above:

- u_date1 and u_date2 are the names of the two date fields. Replace these with your own field names.
- The first line checks that both fields actually have a value.
- The next two lines create variables that have the dates' numerical values.
- The next two lines create different alert messages for the end user: one at the top of the form and one by the u_date1 field in the form.
- The last line aborts the insert or update if the date fields are not correct.

Here is a more complex example of the above comparison. If you have more than one pair of start and end dates, you can use arrays as shown. Additionally, this script requires the input dates to be within a certain range, in this case, no fewer than 30 days in the past and no more than 365 days in the future.

```javascript
// Enter all start and end date fields you wish to check, as well as the previous values
// Make sure that you keep the placement in the sequence the same for all pairs
var startDate = new Array(current.start_date,current.work_start);
var prevStartDate = new Array(previous.start_date,previous.work_start);
var endDate = new Array(current.end_date,current.work_end);
var prevEndDate = new Array(previous.end_date,previous.work_end);

// The text string below is added to the front of 'start must be before end'
var userAlert = new Array('Planned','Work');

// Set the number of Previous Days you want to check
var pd = 30;
// Set the number of Future Days you want to check
var fd = 365;

// You shouldn't have to modify anything below this line
var nowdt = new GlideDateTime();
nowdt.setDisplayValue(gs.nowDateTime());
var nowMs = nowdt.getNumericValue();
var pdms = nowMs;
// Subtract the product of previous days to get value in milliseconds
pdms -= pd * 24 * 60 * 60 * 1000;
var fdms = nowMs;
// Add the product of future days to get value in milliseconds
fdms += fd * 24 * 60 * 60 * 1000;
var badDate = false;

// Iterate through all start and end date / time fields
for (x = 0; x < startDate.length; x ++) {
    if (!startDate[x].nil() && !endDate[x].nil()) {
        var start = startDate[x].getGlideObject().getNumericValue();
        var end = endDate[x].getGlideObject().getNumericValue();
        if (start > end) {
            gs.addInfoMessage(userAlert[x] + ' ' + 'start must be before end');
            startDate[x].setError(userAlert[x] + ' ' + 'start must be before end');
        }
    }
}
```
badDate = true; } 
else if ((prevStartDate[x]) != (startDate[x])) { 
    if (start < pdms) { 
        gs.addInfoMessage(userAlert[x] + ' start must be fewer than ' + pd + ' days ago');
        startDate[x].setError(userAlert[x] + ' start must be fewer than ' + pd + ' days ago');
        badDate = true; } } 
else if ((prevEndDate[x]) != (endDate[x])) { 
    if (end > fdms) { 
        gs.addInfoMessage(userAlert[x] + ' end must be fewer than ' + fd + ' days ahead');
        endDate[x].setError(userAlert[x] + ' end must be fewer than ' + fd + ' days ahead');
        badDate = true; } } 
} if (badDate == true ) { 
current.setAbortAction ( true ); }

Parse XML payloads

Fields in XML format can be parsed with the system's getXMLText function.

Fields that get inserted into the database in XML format, such as the payload of an ecc_event row, can be parsed with the system's getXMLText function. The getXMLText function takes a string and an XPATH expression. For example:

```javascript
var name = gs.getXMLText("<name>joe</name>", "//name");
```

returns the string 'joe'.

Assuming that the field "payload" contains XML, the function call might look like:

```javascript
var name = gs.getXMLText(current.payload, "//name");
```

For information on XPATH, visit w3schools.

Abort a database action in a before business-rule

In a before business rule script, you can cancel or abort the current database action.

In a before business-rule script, you can cancel or abort the current database action using the current.setAbortAction(true) method. For example, if the before business rule is executed during an insert action, and you have a condition in the script that calls current.setAbortAction(true), the new record stored in current is not created in the database.

Determine the operation that triggered the business rule

You can write a script for a business rule that is triggered on more than one database action.

If you want the business rule script to dynamically branch depending on the action that triggered the event, you can use the operation() function. For example:

```javascript
if(current.operation() == "update") { 
current.updates ++; } 
if(current.operation() == "insert") { 
current.updates = 0; }
```

Display business-rules

Display rules are processed when a user requests a record form.
The data is read from the database, display rules are executed, and the form is presented to the user. The current object is available and represents the record retrieved from the database. Any field changes are temporary since they are not yet submitted to the database. To the client, the form values appear to be the values from the database; there is no indication that the values were modified from a display rule. This is a similar concept to calculated fields.

The primary objective of display rules is to use a shared scratchpad object, `g_scratchpad`, which is also sent to the client as part of the form. This can be useful when you need to build client scripts that require server data that is not typically part of the record being displayed. In most cases, this would require a client script making a call back to the server. If the data can be determined prior to the form being displayed, it is more efficient to provide the data to the client on the initial load. The form scratchpad object is an empty object by default, and used only to store name:value pairs of data.

To populate the form scratchpad with data from a display rule:

```javascript
// From display business rule
g_scratchpad.someName = "someValue";
g_scratchpad.anotherName = "anotherValue";

// If you want the client to have access to record fields not being displayed on the form
g_scratchpad.created_by = current.sys_created_by;

// These are simple examples, in most cases you'll probably perform some other queries
// to test or get data
```

To access the form scratchpad data from a client script:

```javascript
// From client script
if(g_scratchpad.someName == "someValue") {
  //do something special
}
```

Use an OR condition in a business rule

An OR condition can be added to any query part within a business rule.

An OR condition can be added to any query part within a business rule with the `addOrCondition()` method. The example below shows a query for finding all the incidents that have either a 1 or a 2 priority. The first `addQuery()` condition is defined as a variable and is used in the OR condition.

```javascript
var inc = new GlideRecord('incident');
var qc = inc.addQuery('priority','1');
qc.addOrCondition('priority','2');
inc.query();
while(inc.next()) {
  // processing for the incident goes here
}
```

The following script is a more complex example, using two query condition variables doing the equivalent of `(priority = 1 OR priority = 2) AND (impact = 2 OR impact = 3)`. The results of the OR condition are run with two variables, `qc1` and `qc2`. This allows you to manipulate the query condition object later in the script, such as inside an IF condition or WHILE loop.

```javascript
var inc = new GlideRecord('incident');
var qc1 = inc.addQuery('priority','1');
qc1.addOrCondition('priority','2');
var qc2 = inc.addQuery('impact','2');
qc2.addOrCondition('impact','3');
```
Reference a Glide list from a business rule

A field defined as a glide list is an array of values stored in a single field.

Here are some examples of how to process a glide_list field when writing business rules. Generally a glide_list field contains a list of reference values to other tables.

Examples

For example, the Watch list field within tasks is a glide_list containing references to user records.

The code below shows how to reference the field.

```javascript
// list will contain a series of reference (sys_id) values separated by a comma
// array will be a javascript array of reference values
var list = current.watch_list.toString();
var array = list.split(",");
for (var i=0; i < array.length; i++) {
    gs.print("Reference value is: " + array[i]);
}
```

You can also get the display values associated with the reference values by using the getDisplayValue() method as shown below.

```javascript
// list will contain a series of display values separated by a comma
// array will be a javascript array of display values
var list = current.watch_list.getDisplayValue();
var array = list.split(",");
for (var i=0; i < array.length; i++) {
    gs.print("Display value is: " + array[i]);
}
```

Use indexOf("searchString") to find a string in a Glide list

Use `indexOf("searchString")` to return the location of the string passed into the method if the glide list field, such as a Watch list, has at least one value in it.

If the field is empty, it returns `undefined`. To avoid returning an undefined value, do any of the following:

- Force the field to a string, such as: `watch_list.toString().indexOf("searchString")`
- Check for an empty Glide list field with a condition before using `indexOf()`, such as: `if (watch_list.nil() || watch_list.indexOf("searchString") == -1)`

Task active state management business rule

A business rule that determines if the active field needs to change based on state changes.

The Task Active State Management business rule is executed for any task state change. It's execution order is 50, designed to run before most other task rules. If the current task table has the close_states attribute defined on it's table or it is inherited from a higher-level table then the rule will determine if the active field needs to change. This is done by comparing the previous and current state values. If the state changes from an active state to an inactive state, the active field will be set to false. If the
state changes from an inactive state to an active state, the active field will be set to true, effectively re-activating/re-opening the task. It is recommended that you leverage this action in your business rules (current.active.changesTo([true/false]) as opposed to creating rules on each task table that marks tasks as inactive or active.

Example script: Locking user accounts

An example script to lock user accounts if the user is not active in the LDAP directory or the user does not have self-service, itil, or admin access to the instance.

The following business rule script locks user accounts if the user is not active in the LDAP directory or the user does not have self-service, itil, or admin access to the instance:

```javascript
// Lock accounts if bcNetIDStatus != 'active' in LDAP and user does not have self-service, itil or admin role
var rls = current.accumulated_roles.toString();
if(current.u_bcnetidstatus == 'active' && (rls.indexOf(',itil,') > 0 || rls.indexOf(',admin,') > 0 || rls.indexOf(',ess,') > 0 )) {
    current.locked_out = false; }
else {
    current.locked_out = true; }

var gr = new GlideRecord("sys_user");
gr.query();
while(gr.next()) {
    gr.update();
    gs.print("updating " + gr.getDisplayValue());
}
```

Example script: default before-query business rule

You can use a query business rule that executes before a database query is made.

You can use a query business rule that executes before a database query is made to prevent users from accessing certain records. Consider the following example from a default business rule that limits access to incident records.

• Name: incident query
• Table: Incident
• When: before, query
• Script:

```javascript
if(!gs.hasRole("itil") && gs.isInteractive()) {
    var u = gs.getUserID();
    var qc =
        current.addQuery("caller_id",u).addOrCondition("opened_by",u).addOrCondition("watch_list","CONTAINS",u);
    gs.print("query restricted to user: " + u);
}
```

This example prevents users from accessing incident records unless they have the itil role are listed in the Caller or Opened by field. So, for example, when self-service users open a list of incidents, they can only see the incidents they submitted.

**Note:** You can also use access controls to restrict the records that users can see.

Script includes

Script includes are used to store JavaScript that runs on the server.
Create script includes to store JavaScript functions and classes for use by server scripts. Each script include defines either an object class or a function.

Consider using script includes instead of global business rules because script includes are only loaded on request.

**Script include form**

Script includes have a name, description and script. They also specify whether they are active or not, and whether they can be called from a client script.

To access script includes, navigate to System Definitions > Script Includes.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the script include. If you are defining a class, this must match the name of the class, prototype, and type. If you are using a classless (on-demand) script include, the name must match the function name.</td>
</tr>
<tr>
<td>Client callable</td>
<td>Makes the script include available to client scripts, list/report filters, reference qualifiers, or if specified as part of the URL.</td>
</tr>
<tr>
<td>Application</td>
<td>The application where this script include resides.</td>
</tr>
<tr>
<td>Accessible from</td>
<td>Sets which applications can access this script include:</td>
</tr>
<tr>
<td></td>
<td><strong>All application scopes</strong> Can be accessed from any application scope.</td>
</tr>
<tr>
<td></td>
<td><strong>This application scope only</strong> Can be accessed only from the current application scope.</td>
</tr>
<tr>
<td></td>
<td>See Scripting in Scoped Applications for information on scoped scripts.</td>
</tr>
<tr>
<td>Active</td>
<td>Enables the script include when selected. Uncheck the active field to disable the script include.</td>
</tr>
<tr>
<td>Description</td>
<td>Provides descriptive content regarding the script include.</td>
</tr>
<tr>
<td>Script</td>
<td>Defines the server side script to run when called from other scripts. The script must define a single JavaScript class or a global function. The class or function name must match the Name field.</td>
</tr>
<tr>
<td>Package</td>
<td>The package that contains this script include.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Created by</td>
<td>The user who created this script include.</td>
</tr>
<tr>
<td>Updated by</td>
<td>The user who most recently updated this script include.</td>
</tr>
<tr>
<td>Protection policy</td>
<td>Sets the level of protection for the script include:</td>
</tr>
<tr>
<td>None</td>
<td>Allows anyone to read and edit this downloaded or installed script include.</td>
</tr>
<tr>
<td>Read-only</td>
<td>Allows anyone to read values from this downloaded or installed script include. No one can change script values on the instance on which they download or install the script include.</td>
</tr>
<tr>
<td>Protected</td>
<td>Provides intellectual property protection for application developers.</td>
</tr>
<tr>
<td>Related lists on the form view:</td>
<td></td>
</tr>
<tr>
<td>Versions</td>
<td>Shows all versions of the script include. Use this list to compare versions or to revert to a previous version.</td>
</tr>
</tbody>
</table>
type. When you create a new script include and give it a name, the system provides you a code snippet with the class and prototype set up properly.

```javascript
var NewInclude = Class.create();
NewInclude.prototype = {
    initialize: function() {},
    myFunction: function() {//Put function code here},
    type: 'NewInclude'};
```

You could then use the 'myFunction' line like this:

```javascript
var foo = new NewInclude();
foo.myFunction();
```

**Note:** Try not to modify a ServiceNow supplied script include. If you want a script include that does something similar to an existing one, copy it and make changes to the copy or consider extending the object. This is a common practice when using GlideAjax.

### Privacy settings

The privacy setting for a client-callable script-include can be public or private.

The private privacy-setting means that guests who access public pages cannot access the client-callable script-include. A private script cannot be executed by a non-logged-in user.

A public privacy-setting means that the client script can be executed by non-logged-in users that create an appropriate HTTP request. This can create a security problem if the client script provides confidential information.

The following script includes remain public by default because public pages need to access them:

- GlideSystemAjax
- SysMessageAjax
- KnowledgeMessagingAjax
- KnowledgeAjax
- PasswordResetAjax

### Change privacy on all client-callable script includes

How to change the privacy on all client-callable script includes.

To provide further control over all client-callable script includes, administrators can add the property glide.script.ccsi.ispublic. This property changes the visibility of client-callable script includes by making them all public or private. Configure the property as follows:

<table>
<thead>
<tr>
<th>Title</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>glide.script.ccsi.ispublic</td>
</tr>
<tr>
<td>Type</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>false</td>
</tr>
</tbody>
</table>
Change privacy on a single client callable script include

Change the privacy setting for a single client-callable script-include by adding the isPublic() function. The isPublic() setting takes precedence over the glide.script.ccsi.ispublic property. For example, if the property is set to false making all client-callable script-includes private, and a script sets isPublic() to true, the script is public.

To change the privacy for a single client-callable script include, add the following method to the script include:

```javascript
isPublic:function(){return[true/false];},
```

Use either true or false for the script include.

Script actions

You can use script actions to create server-side scripts that perform a variety of tasks, such as modifying a configuration item (CI), or managing failed login attempts. Script actions are triggered by events only.
### Configuration

To create a new script action, navigate to **System Policy > Events > Script Actions** and click **New**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a unique name for your script action.</td>
</tr>
<tr>
<td>Event Name</td>
<td>Select the event to use for this script. If you do not find an event for your script action that suits your purpose, you can create a new one.</td>
</tr>
<tr>
<td>Application</td>
<td>The application that contains this script.</td>
</tr>
<tr>
<td>Order</td>
<td>The order in which the script will be executed.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box (true) to enable this script action.</td>
</tr>
<tr>
<td>Condition Script</td>
<td>Create a statement for a condition under which this script should execute. The system only parses the <strong>Script</strong> field if the condition evaluates to true. If you decide to include the condition statement in the script, leave this field blank.</td>
</tr>
<tr>
<td>Script</td>
<td>Create a script that runs when the condition you define evaluates to true. Two additional objects are available in this script:</td>
</tr>
<tr>
<td></td>
<td>- <strong>event</strong>: a GlideRecord - the sysevent that caused this to be invoked. If you wanted so get this first parameter on the event, you would use event.parm1 or event.parm2 for the second parameter. For the date/time of the event, use event.sys_created_on. To get the user ID that created the event (if there was a user associated), use event.user_id.</td>
</tr>
<tr>
<td></td>
<td>- <strong>current</strong>: a GlideRecord - the event scheduled on behalf of (incident for example).</td>
</tr>
</tbody>
</table>

This is a sample of a script action that creates an email notification for Workflow activity:
```javascript
/*
  * Handle a workflow.notification event by creating an email notification that can be sent
  * parem1 - sys_id of workflow activity, sys_id of workflow context
  * parem2 - comma-separated list of recipients
  */

sendWorkflowNotification();

function sendWorkflowNotification() {

  var activity = new GlideRecord('wf_activity');
  if (!activity.get(parem1))
    return 0;

  var context = new GlideRecord('wf_context');
  if (parem2.length == 2)
    context.put(parem2);

  GlideController.putGlobal('context', context);
  GlideController.putGlobal('workflow', workflow);

  var subject = workflow(activity.vars,subject);
  var message = workflow(activity.vars,mail);
  var emailAction = new GlideEmailAction();

  var email = new GlideRecord('sysevent_email_action');
```
UI pages

UI pages can be used to create and display forms, dialogs, lists and other UI components.

Use UI pages as widgets on homepages. To find the UI pages, navigate to System UI > UI Pages.

This functionality requires a knowledge of HTML or Jelly.
Figure 937: UI page form

Client script:

```javascript
var t = gc("engines");
var tbody = t.firstChild;
initEngines();

function initEngines()
{
    var tables = [];
    for (var i = 0; i < t.engines.length; i++)
    {
        var table = tables.split(",");
        for (var i = 0; i < table.length; i++)
        {
            var table = tableList[i];
            var key = "glide.approval_engine." + table;
            var nvalue = gs.getParameter('engine_' + table) + ";
            if (nvalue != "nvalue")
            {
                gs.print("Set approval engine property " + key + " to " + nvalue + ");
            }
        }
    }
}
```

Processing script:

```javascript
var tableList = tables.split(",");
for (var i = 0; i < tableList.length; i++)
{
    var table = tableList[i];
    var key = "glide.approval_engine." + table;
    var nvalue = gs.getProperty(key) + ";
    var nvalue = request.getParameter('engine_' + table) + ";
    if (nvalue != "nvalue")
    {
        gs.print("Set approval engine property " + key + " to " + nvalue + ");
    }
}
```

Update Try It Delete

Compare this version of the UI Page with previous versions.

© 2017 ServiceNow. All rights reserved.
The UI page form provides the following fields:

### Table 1361: UI page

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Name used to invoke the page via a URL (must not contain spaces).</td>
</tr>
<tr>
<td><strong>HTML</strong></td>
<td>Main component of the page, and it defines what will be rendered when the page is shown. It can contain either static XHTML or dynamically generated content defined as Jelly, and it can call script includes and UI Macros.</td>
</tr>
<tr>
<td><strong>Client Script</strong></td>
<td>Client-side JavaScript that runs in the browser (e.g. functions called by buttons, etc.). It is intended to handle any client-side processing needed, for example setting focus to a field, or other interactive DHTML features after a page is loaded. Ultimately, a UI page's Client Scripts are deployed to the browser within a <code>&lt;script/&gt;</code> tag, so it could be defined within the page's HTML field to achieve the same effect. Using the Client Script field instead to define these scripts makes things much more tidy and readable though, and it keeps the Jelly and HTML from becoming unmanageable.</td>
</tr>
<tr>
<td><strong>Processing Script</strong></td>
<td>Script that runs on the server when the page is submitted. This is useful if your page has a form (defined with the <code>&lt;g:ui_form/&gt;</code> or <code>&lt;g:form/&gt;</code> tags).</td>
</tr>
</tbody>
</table>

**Related lists on the form view:**

- **Versions**
  - Shows all versions of the UI page.

A UI page can be secured by creating an ACL with the following parameters:

- **Type**: `ui_page`
- **Operation**: `read`
- **Name**: name of the UI page to be protected

For details on creating an ACL rule, see [Create an ACL rule](#).

### UI page access

Each UI page has a URL computed from the application scope, page name, and the `.do` file extension.

For example, to display the page called `glidewindow_example` on the demo system, you would navigate to `https://<instance name>.service-now.com/glidewindow_example.do`. If the page was part of a custom application called `example_app`, you would instead navigate to `https://<instance name>.service-now.com/x_example_app_glidewindow_example.do`. 
You can also add additional parameters to a URL that can be accessed within a page's HTML section as jelly variables. That is, appending arguments to the URL as follows: `/my_test_page.do?sysparm_verbose=true` creates jelly variables called `verbose` that can be accessed as follows:

```plaintext
<j2:if test="(!$[!empty(sysparm_verbose)])"> <span>show extra stuff </span> </j2:if>
```

A common practical example of this might be retrieving a database record for display. To build a list of a user's roles, pass in a parameter with the user's `sys_id`. Invoke the following UI page to display a list of roles for that user with Jelly code:

```plaintext
role_select.do?sysparm_user=5137153cc611227c000b7bd81bd82d07
```

```plaintext
<j:set var="jvar_user_id" value="${sysparm_user}" />

<g:evaluate> var userRoles = new GlideRecord('sys_user_has_role');
userRoles.addQuery('user','${jvar_user_id}');
userRoles.query();
</g:evaluate>

<select id='select_role'>
<j:while test="${userRoles.next()}">
  <option value="${userRoles.sys_id}">${userRoles.role.name} </option>
</j:while>
</select>
```

An exception to be careful of, though, is the reserved variable name `sys_id`. This variable always contains the ID of the UI page itself, regardless of what is specified in the URL. A common substitute variable name is `sysparm_id`.

### UI page process scripts

If your UI page contains a form (uses the `<g:form>` tag), you can submit the form and have the process script run.

The processing script can naturally access fields on the form. For example, if your form contained the `application_sys_id` field:

```plaintext
<g:ui_form><p>Click OK to run the processing script.</p>
<g:dialog_buttons_ok_cancel ok = "return true"/> <input type = "hidden"
name = "application_sys_id" value = "499836460a0a0b1700003e7ad950b5da"/></g:ui_form>
```

You could then access it using just `application_sys_id`:

```
var application = new GlideRecord('hr_application');
application.get(application_sys_id);
application.status = "Rejected";
application.update();
var urlOnStack = GlideSession.get().getStack().bottom();
response.sendRedirect(urlOnStack);
```

**Note:**
- `GlideSession` replaces `Packages.com.glide.sys.GlideSession`
If you are using the UI page for a dialog, you can also reference the most recent URL on the stack using the code above and then send the response to that location. This is useful if you want to have the dialog’s processing script update something and then redisplay the screen that brought up the dialog.

**UI Macros**

UI macros define modular, reusable components in the ServiceNow platform. For example, UI Macros are used to apply formatters to various forms within ServiceNow that provide additional information about the form without using a field. This functionality requires the knowledge of Jelly script.

To administer UI macros, navigate to **System UI > UI Macros** and define the following information:

- **Name**: A unique and descriptive name for this macro.
- **Active**: Select the check box to render the element as defined. Clear the check box to disable the element without deleting the code. For example, the `email_reply` macro is inactive by default.
- **Description**: Describe the purpose of the macro and parameters passed to it.
- **XML**: Jelly script that defines the macro.

**Examples**

UI Macros can invoke other UI Macros. In this example, the UI Macro invokes the `ui_update_set_picker` and `ui_timezone_changer` macros:

```xml
<?xml version= "1.0" encoding="utf-8"?>
<j:jelly trim= "false" xmlns:j= "jelly:core" xmlns:g= "glide" xmlns:j2= "null" xmlns:g2= "null">
  <g:ui_update_set_picker />
  <g:ui_timezone_changer />
</j:jelly>
```

UI Macros can be invoked from UI Pages:

```xml
<g:macro_invoke macro="decorate_welcome_header_stripe_left" />
```

**Processors**

Processors are analogous to Java servlets. Processors provide a customizable URL endpoint that can execute arbitrary server-side JavaScript code and produce output such as TEXT, JSON, or HTML.

Typically, you create custom processors when you want a URL query to:

- Produce data in non-standard formats.
- Perform non-standard record operations.
- Act on multiple tables.

There are two types of processors.

- **Single-purpose**: processors that have a single purpose and single endpoint.
- **Multi-table**: processors that apply to all tables on the instance.

By default, processors are public and can execute without requiring authorization. This means that any logged-in user can execute the processor without needing a particular role. To prevent leakage of secure data, protect your processors by requiring a CSRF token or requiring users have an authorized role to run the processor.
Processor form
The processor form has the following fields.

### Table 1362: Processor Form

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unique name of the processor.</td>
</tr>
<tr>
<td>Type</td>
<td>Programming language of the processor script. Options include:</td>
</tr>
<tr>
<td></td>
<td>- java: do not select this option</td>
</tr>
<tr>
<td></td>
<td>- script</td>
</tr>
<tr>
<td>Application</td>
<td>Application containing this record.</td>
</tr>
<tr>
<td>Active</td>
<td>Flag to enable or disable the record.</td>
</tr>
<tr>
<td>CSRF protect</td>
<td>Option to protect the processor from running unless the instance uses a CSRF token.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the processor's function or purpose.</td>
</tr>
<tr>
<td>Parameters</td>
<td>List of available input parameters. Specify parameter values in the URL as:&lt;parameter name&gt;=&lt;parameter value&gt;.</td>
</tr>
<tr>
<td>Path</td>
<td>URI path used to call this processor. Call a processor from the URL as:</td>
</tr>
<tr>
<td></td>
<td>https://&lt;instance name&gt;.service-now.com/&lt;Path&gt;.do</td>
</tr>
<tr>
<td>Script</td>
<td>Immediately Invoked Function Expression to run when the system calls this processor.</td>
</tr>
<tr>
<td></td>
<td>The function automatically provides input parameters for the following API objects.</td>
</tr>
<tr>
<td></td>
<td>- g_request</td>
</tr>
<tr>
<td></td>
<td>- g_response</td>
</tr>
<tr>
<td></td>
<td>- g_processor</td>
</tr>
<tr>
<td>Protection policy</td>
<td>Policy to use to protect this record's script. Options include:</td>
</tr>
<tr>
<td></td>
<td>- None</td>
</tr>
<tr>
<td></td>
<td>- Read-only</td>
</tr>
<tr>
<td></td>
<td>- Protected</td>
</tr>
</tbody>
</table>
Processor API components

Processors have access to dedicated API classes, objects, and methods.

### Table 1363: Processor API components

<table>
<thead>
<tr>
<th>Class, object, or method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_response</td>
<td>An object of type HttpServletResponse. See Interface HttpServletResponse.</td>
</tr>
<tr>
<td><code>setContentType('text/html; charset=UTF-8')</code></td>
<td>A HttpServletResponse method to set the content type of the response being sent to the client. See Interface HttpServletResponse.</td>
</tr>
<tr>
<td>g_request</td>
<td>An object of type HttpServletRequest. See Interface HttpServletRequest.</td>
</tr>
<tr>
<td><code>getParameter()</code></td>
<td>A glide method to get the value of a URL parameter.</td>
</tr>
<tr>
<td>canRead()</td>
<td>A GlideRecord method to determine if the user can read data from a table. See Scoped GlideRecord API Reference.</td>
</tr>
<tr>
<td>g_processor</td>
<td>A simplified servlet for processors.</td>
</tr>
<tr>
<td><code>writeOutput()</code></td>
<td>A processor method to display information on the client.</td>
</tr>
<tr>
<td>g_target</td>
<td>An object containing the target table name of a processor URL. For example, a processor containing the URI <code>incident.do</code> applies to the Incident table.</td>
</tr>
</tbody>
</table>

### Secure a processor

You can protect your processor against unauthorized use by using role restrictions.

You can re-use a table's user role restrictions to protect it from access by your processor. This protection method assumes the processor will access table data.

1. Create or select a user role that has access to the table the processor script calls.
2. Navigate to System Definition > Processors.
3. In Script, add the following code block.

   ```javascript
   var gr = new GlideRecord('your_table_name');
   // canRead() compares the table’s ACL to the user making this request,
   // and returns true if the logged-in user has read access to this table
   if(gr.canRead())
   {
     // Perform table query here
     g_processor.writeOutput('Success!');
   }
   else {
     g_processor.writeOutput('You do not have permission to read table your_table_name');
   }
   ```

4. Update the code block to use other access restrictions as needed.

Available access functions include:
Create a simple processor

In this exercise, you will create a simple processor to execute a script from a URL query.

The following steps assume that you have your own demonstration instance.

1. Navigate to System Definition > Processors.
2. Click New.
3. Enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Hello</td>
</tr>
<tr>
<td>Type</td>
<td>Script</td>
</tr>
<tr>
<td>Path</td>
<td>Hello</td>
</tr>
<tr>
<td>Script</td>
<td>var name= g_request.getParameter(&quot;name&quot;); g_processor.writeOutput(&quot;text/plain&quot;,&quot;Hello &quot;+name);</td>
</tr>
</tbody>
</table>

4. Click Submit.
5. Enter a URL query to the instance with the following format: https://instance.service-now.com/processor_name.do?parameter=value. For example: https://<instancename>.service-now.com/Hello.do?name=world.

Create a multi-table processor

In this exercise, you will create a multi-table processor that reports the number of rows in any table on your instance.

The multi-table processor protects itself from performance and security violations by confirming that the user is authorized to read the table. It does not report on certain tables that are too large to query safely.

1. Navigate to System Definition > Processors.
2. Click New.
3. Enter the following information:

<table>
<thead>
<tr>
<th>Name</th>
<th>TableSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Choose Javascript</td>
</tr>
<tr>
<td>Description</td>
<td>Return number of records in a table</td>
</tr>
<tr>
<td>Parameters</td>
<td>SIZE</td>
</tr>
<tr>
<td>Path</td>
<td>&lt;leave empty&gt;</td>
</tr>
</tbody>
</table>
Script

```javascript
var name = g_request.getParameter("name");
g_processor.writeOutput("text/plain", "Hello " + name);
```

Create a custom processor

You can create a custom processor to execute a script from a URL query.

When complete, you will be able to:

- Create a new processor
- Run a script from a URL query

The following example steps assume you have your own demonstration instance.

1. Navigate to **System Definition > Processors**.
2. Click **New**.
3. For **Name**, enter **Hello**.
4. For **Type**, select **script**.
5. For **Path**, enter **Hello**.
6. For **Script**, enter the following code.

```
var name = g_request.getParameter("name");
g_processor.writeOutput("text/plain", "Hello " + name);
```
7. Click **Submit**.
8. Logout of the instance and open a new browser window.
9. Enter a URL query to the instance with the following format: `https://instance.service-now.com/processor_name.do?parameter=value`. For example: `https://<instancename>.service-now.com/Hello.do?name=world`.
10. When prompted, enter credentials for valid user.

Installation settings

Installation Settings permit the programmatic determination of a setting.

Installation settings are global business rules with calculated names. Installation settings are calculated just before a record is displayed and therefore facilitate very dynamic determination of access and roles. Installation settings controlling access to fields and records are:

- CanRead()
- CanWrite()
- CanCreate()
- CanDelete()

Functions can return true if access is permitted, false if not. No return value uses the permission calculated using roles. The function has access to the current record through the variable current code.

The name of the function checking the permission on a record is formed by prefixing the setting name with the record name:

record_nameCanRead()

Similarly, the permission on a field in a record are formed by prefixing the function name with the record name, underbar (_) and field name:

record_name_field_nameCanRead()

Naming Examples:

function incidentCanWrite() {} // can user write to this record?
function incident_numberCanWrite() {}  // can user write to the number field?

This sample business rule restricts the writing of the name field in the sys_dictionary file when the entry already exists:

// the element name cannot be written unless this is a new record (not yet in database)
function sys_dictionary_nameCanWrite() {
    if (current.isNewRecord())
        return;
    return false;
}

Using DurationCalculator to calculate a due date

Using the DurationCalculator script include, you can calculate the due date, using either a simple duration or relative duration.

The following script demonstrates how to use DurationCalculator to calculate a due date. The first part of the script illustrates how to set a start datetime using the setStartDateTime() method and then use the calcDuration() method to determine a due date that is "x" amount of continuous time (seconds) from the specified start datetime. The second half of the script illustrates how to use DurationCalculator to calculate a due date based on a schedule. Schedules enable you to apply a “filter” on future time, such as only including the days in a work week within the calculation. For example, if you apply a schedule
"weekdays" (which only includes Monday through Friday) to your duration calculation, and the start
datetime is Friday at 5:00 pm, when you add a duration of two days, your due date would be Tuesday
at 5:00 pm. If you did not use a schedule, your due date would be Sunday at 5:00 pm. For additional
information on schedules, see Schedules on page 1040.

This script can be cut and pasted into the Scripts Background page and run as is. It can also serve as an
example for authoring business rules, UI actions, or used any other place that server-side script can be
authored.

```javascript
/**
 * Demonstrate the use of DurationCalculator to compute a due date.
 *
 * You must have a start date and a duration. Then you can compute a
 * due date using the constraints of a schedule.
 */

gs.include('DurationCalculator');
executeSample();

/**
 * Function to house the sample script.
 */
function executeSample(){

    // First we need a DurationCalculator object.
    var dc = new DurationCalculator();

    // -------------- No schedule examples --------------

    // Simple computation of a due date without using a schedule. Seconds
    // are added to the start date continuously to get to a due date.
    var gdt = new GlideDateTime("2012-05-01 00:00:00");
    dc.setStartDateTime(gdt);
    if(!dc.calcDuration(2*24*3600)){ // 2 days
        gs.log("*** Error calculating duration");
        return;
    }
    gs.log("calcDuration no schedule: "+ dc.getEndDateTime()); // "2012-05-03 00:00:00" two days later

    // Start in the middle of the night (2:00 am) and compute a due date 1
    // hour in the future.
    // Without a schedule this yields 3:00 am.
    var gdt = new GlideDateTime("2012-05-03 02:00:00");
    dc.setStartDateTime(gdt);
    if(!dc.calcDuration(3600)){
        gs.log("*** Error calculating duration");
        return;
    }
    gs.log("Middle of night + 1 hour (no schedule): "+
    dc.getEndDateTime()); // No scheduled start date, just add 1 hour

    // -------------- Add a schedule to the date calculator

    addSchedule(dc);

    // Start in the middle of the night and compute a due date 1 hour in the
    future.
    // Since we start at 2:00 am the computation adds the 1 hour from the
    start
    // of the day, 8:00am to get to 9:00am
    var gdt = new GlideDateTime("2012-05-03 02:00:00");
```

© 2017 ServiceNow. All rights reserved. 3879
dc.setStartDateTime(gdt);
if(!dc.calcDuration(3600)) {
    gs.log("*** Error calculating duration");
    return;
}
gs.log("Middle of night + 1 hour (with 8-5 schedule): " +
dc.getEndDateTime()); // 9:00 am

// Start in the afternoon and add hours beyond quitting time. Our
// schedule says the work day
// ends at 5:00pm, if the duration extends beyond that, we roll over to
the next work day.
// In this example we are adding 4 hours to 3:00pm which gives us 10:00
am the next day.
var gdt = new GlideDateTime("2012-05-03 15:00:00");
dc.setStartDateTime(gdt);
if(!dc.calcDuration(4*3600)) {
    gs.log("*** Error calculating duration");
    return;
} gs.log("Afternoon + 4 hour (with 8-5 schedule): " +
dc.getEndDateTime()); // 10:00 am.

// This is a demo of adding 2 hours repeatedly and examine the result.
This
// is a good way to visualize the result of a due date calculation.
var gdt = new GlideDateTime("2012-05-03 15:00:00");
dc.setStartDateTime(gdt);
for(var i=2; i<24; i+=1) {
    if(!dc.calcDuration(i*3600)) {
        gs.log("*** Error calculating duration");
        return;
    }
gs.log("add " + i +" hours gives due date: " + dc.getEndDateTime());
}

// Setting the timezone causes the schedule to be interpreted in the
specified timezone.
// Run the same code as above with different timezone. Note that the 8
to 5 workday is
// offset by the two hours as specified in our timezone.
dc.setTimeZone("GMT-2");
var gdt = new GlideDateTime("2012-05-03 15:00:00");
dc.setStartDateTime(gdt);
for(var i=2; i<24; i+=1) {
    if(!dc.calcDuration(i*3600)) {
        gs.log("*** Error calculating duration");
        return;
    }
gs.log("add " + i +" hours gives due date (GMT-2): " +
dc.getEndDateTime());
}

/**
 * Add a specific schedule to the DurationCalculator object.
 * @param durationCalculator An instance of DurationCalculator
 */
function addSchedule(durationCalculator) {
    // Load the "8-5 weekdays excluding holidays" schedule into our duration
calculator.
    var scheduleName = "8-5 weekdays excluding holidays";
    var grSched = new GlideRecord('cmn_schedule');
Using DurationCalculator to compute a simple duration

A simple duration is the number of seconds between two date times.

If no schedule is used then this is a simple time date subtraction. If a schedule is used, then the schedule is consulted to remove non-work hours from the computation. Suppose schedule "8-5 weekdays excluding holidays" is used. In this case, the number of work hours from noon Monday to noon Tuesday is nine hours. To compute a simple duration, initialize DurationCalculator and call the `calcScheduleDuration()` method.

This script demonstrates how to use DurationCalculator to compute a simple duration.

```java
/**
 * Sample script demonstrating use of DurationCalculator to compute simple durations
 */
gs.include('DurationCalculator');
executeSample();

/**
 * Function to house the sample script.
 */
function executeSample(){
    // First we need a DurationCalculator object.
    var dc = new DurationCalculator();

    // Compute a simple duration without any schedule. The arguments 
    // can also be of type GlideDateTime, such as fields from a GlideRecord.
    var dur = dc.calcScheduleDuration("2012-05-01", "2012-05-02");
    gs.log("calcScheduleDuration no schedule: " + dur); // 86400 seconds (24 hours)

    // The above sample is useful in limited cases. We almost always want to
    // use some schedule in a duration computation, let's load a schedule.
    addSchedule(dc);

    // Compute a duration using the schedule. The schedule 
    // specifies a nine hour work day. The output of this is 32400 seconds, 
    // or 
    // a nine hour span.
    dur = dc.calcScheduleDuration("2012-05-23 12:00:00","2012-05-24 
    12:00:00");
    gs.log("calcScheduleDuration with schedule: " + dur); // 32400 seconds (9 hours)

    // Compute a duration that spans a weekend and holiday. Even though this
    // spans three days, it only spans 9 work hours based on the schedule.
}
```

See the ServiceNow Developers site for API information.
dur = dc.calcScheduleDuration("2012-05-25 12:00:00", "2012-05-29 12:00:00");
gs.log("calcScheduleDuration with schedule spanning holiday: " + dur); // 32400 seconds (9 hours)

// Use the current date time in a calculation. The output of this is dependent on when you run it.
var now = new Date();
dur = dc.calcScheduleDuration("2012-05-15", new GlideDateTime());
gs.log("calcScheduleDuration with schedule to now: " + dur); // Different on every run.

/**
 * Add a specific schedule to the DurationCalculator object.
 *
 * @param durationCalculator An instance of DurationCalculator
 */
function addSchedule(durationCalculator){
   // Load the "8-5 weekdays excluding holidays" schedule into our duration calculator.
   var scheduleName = "8-5 weekdays excluding holidays";
   var grSched = new GlideRecord('cmn_schedule');
grSched.addQuery('name', scheduleName);
grSched.query();
if(!grSched.next()){
   gs.log('*** Could not find schedule "' + scheduleName + '"");
   return;
}
durationCalculator.setSchedule(grSched.getUniqueValue());
}

Using relative duration

Relative duration is very similar to simple duration except a piece of script is used to determine what parts of a day to remove from the difference calculation.

This script is stored in table cmn_relative_duration and can be examined by navigating to System Scheduler > Schedules > Relative Durations. There are some example relative duration scripts in the out-of-the-box instance.

A relative duration sys_id is passed to the method calcRelativeDuration() of the DurationCalculator class after initialization. When this method is called, the DurationCalculator object is passed to the relative duration script (stored in table cmn_relative_duration) as the variable calculator. So, the relative duration script you write and store in cmn_relative_duration has access to the executing DurationCalculator through the variable calculator.

The following script demonstrates how to use DurationCalculator to calculate a relative duration.

/**
 * Sample use of relative duration calculation.
 *
 */
gs.include('DurationCalculator');
executeSample();

/**
 * Function to house the sample script.
 */
function executeSample(){
   // First we need a DurationCalculator object. We will also use
```javascript
// the out-of-box relative duration "2 bus days by 4pm"
var dc = new DurationCalculator();
var relDur = "3bf802c20a0a0b52008e2859cd8abcf2"; // 2 bus days by 4pm if before 10am
addSchedule(dc);

// Since our start date is before 10:00am our result is two days from now at 4:00pm.
var gdt = new GlideDateTime("2012-05-01 09:00:00");
dc.setStartDateTime(gdt);
if(!dc.calcRelativeDuration(relDur)){
gs.log("*** calcRelativeDuration failed");
return;
}
gs.log("Two days later 4:00pm: "+ dc.getEndDateTime());

// Since our start date is after 10:00am our result is three days from now at 4:00pm.
var gdt = new GlideDateTime("2012-05-01 11:00:00");
dc.setStartDateTime(gdt);
if(!dc.calcRelativeDuration(relDur)){
gs.log("*** calcRelativeDuration failed");
return;
}
gs.log("Three days later 4:00pm: "+ dc.getEndDateTime());

/**
 * Add a specific schedule to the DurationCalculator object.
 * @param durationCalculator An instance of DurationCalculator
 */
function addSchedule(durationCalculator){
    // Load the "8-5 weekdays excluding holidays" schedule into our duration calculator.
    var scheduleName ="8-5 weekdays excluding holidays";
    var grSched =new GlideRecord('cmn_schedule');
grSched.addQuery('name', scheduleName);
grSched.query();
if(!grSched.next()){ gs.log("*** Could not find schedule "+ scheduleName +""");
return;
}
durationCalculator.setSchedule(grSched.getUniqueValue(),"GMT");
}
```}

**Using GlideRecord to query tables**

In order to query a table, first create an object for the table.

This object is called a GlideRecord. To create a GlideRecord, create the following in script:

```javascript
var target = new GlideRecord('incident');
```

This creates a variable called target which is a GlideRecord object for the incident table. The only parameter needed is the name of the table to be accessed.

To process all records from the incident table, add the following script:

```javascript
target.query(); // Issue the query to the database to get all records
while (target.next()) {
    // add code here to process the incident record
}
```
This issues the `query()` to the database. Each call to `next()` would load the next record which you would process and do whatever you want to do.

But that is not the common case. Most of the time you actually want to retrieve a specific record or a specific set of records, and you have some criteria (query conditions) that define the records you want to obtain. For example, say you want to obtain all the incident records that have a priority value of 1. Here is the code that would accomplish that.

```javascript
var target = new GlideRecord('incident');
target.addQuery('priority', 1);
target.query(); // Issue the query to the database to get relevant records
while (target.next()) {
    // add code here to process the incident record
}
```

Notice in the above code we added the line `target.addQuery('priority', 1);`. This is indicating that you only want the records where the priority field is equal to 1. We assume that the majority of queries that you will want to do will be equality queries, queries where you want to find records where a field is equal to a value. Therefore we provide this format of the query and do not make you specify that you want an equals operation, we just assume it. However, lets say you wanted to find all incidents where the priority field is GREATER THAN 1. In that case you have to provide us with the operator that you want to apply to priority and this is done by providing the operator in the `addQuery()` request as is shown below.

```javascript
var target = new GlideRecord('incident');
target.addQuery('priority', '>', 1);
target.query(); // Issue the query to the database to get relevant records
while (target.next()) {
    // add code here to process the incident record
}
```

### Available JavaScript operators

The table below contains the operators that can be supplied to the `addQuery()` request.

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
<th>addQuery</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>=</code></td>
<td>Field must be equal to value supplied.</td>
<td><code>addQuery('priority', '=', 1);</code></td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>Field must be greater than value supplied.</td>
<td><code>addQuery('priority', '&gt;', 1);</code></td>
</tr>
<tr>
<td><code>&lt;</code></td>
<td>Field must be less than value supplied.</td>
<td><code>addQuery('priority', '&lt;', 3);</code></td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Field must be equal or greater than value supplied.</td>
<td><code>addQuery('priority', '&gt;=', 1);</code></td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>Field must be equal or less than value supplied.</td>
<td><code>addQuery('priority', '&lt;=', 3);</code></td>
</tr>
<tr>
<td><code>!=</code></td>
<td>Field must not equal the value supplied.</td>
<td><code>addQuery('priority', '!=', 1);</code></td>
</tr>
<tr>
<td>Field</td>
<td>Definition</td>
<td>addQuery</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STARTSWITH</td>
<td>Field must start with the value supplied. The example shown on the right will get all records where the <em>short_description</em> field starts with the text Error.</td>
<td>addQuery('short_description', 'STARTSWITH', 'Error');</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>Field must contain the value supplied somewhere in the text. The example shown on the right will get all records where the <em>short_description</em> field contains the text Error anywhere in the field.</td>
<td>addQuery('short_description', 'CONTAINS', 'Error');</td>
</tr>
<tr>
<td>IN</td>
<td>Takes a map of values that allows commas, and gathers a collection of records that meet some other requirement. Behaves as <code>Select * from &lt;table&gt; where short_description IN ('Error'), which is identical to </code>Select * from &lt;table&gt; where short_description='Error'. For example, to query all variable values that belong to a specific Activity, use the IN clause to query all Activities that are of that type, and store their sys_ids in a map, or comma-separated list. Then query the variable value table and supply this list of sys_ids.</td>
<td>addQuery('short_description', 'IN', 'Error,Success,Failure');</td>
</tr>
<tr>
<td>ENDSWITH</td>
<td>Field must terminate with the value supplied. The example shown on the right will get all records where the <em>short_description</em> field ends with text Error.</td>
<td>addQuery('short_description', 'ENDSWITH', 'Error');</td>
</tr>
<tr>
<td>DOES NOT CONTAIN</td>
<td>Selects records that do NOT match the pattern in the field. This operator does not retrieve empty fields. For empty values, use the operators &quot;is empty&quot; or &quot;is not empty&quot;. The example shown on the right will get all records where the <em>short_description</em> field does not have the word &quot;Error&quot;.</td>
<td>addQuery('short_description', 'DOES NOT CONTAIN', 'Error');</td>
</tr>
</tbody>
</table>
### Field: NOT IN

**Definition:**
Takes a map of values that allows commas, and gathers a collection of records that meet some other requirement. Behaves as: Select * from <table> where short_description NOT IN ('Error').

**addQuery**
```
addQuery('short_description', 'NOT IN', 'Error,Success,Failure');
```

**Description:**
Special operator that allows you to retrieve only records of a specified "class" for tables which are extended. For example when going after configuration items (cmdb_ci table) you many want to retrieve all configuration items that are have are classified as computers. The code to the right will do that.

**addQuery**
```
addQuery('sys_class_name', 'INSTANCEOF', 'cmdb_ci_computer');
```

---

There are also some special methods that can be used when you want to search for data that is NULL or NOT NULL. To search for all incidents where the `short_description` field has not been supplied (is null), use the following query:

```javascript
var target = new GlideRecord('incident');
target.addNullQuery('short_description');
target.query(); // Issue the query to the database to get all records
while (target.next()) {
    // add code here to process the incident record
}
```

To find all incidents in which a `short_description` has been supplied, use the following query:

```javascript
var target = new GlideRecord('incident');
target.addNotNullQuery('short_description');
target.query(); // Issue the query to the database to get all records
while (target.next()) {
    // add code here to process the incident record
}
```

For additional information see [GlideRecord()](#) on the ServiceNow Developer Site.

## GlideRecord query examples

These examples demonstrate how to perform various GlideRecord queries.

### query

```javascript
var rec = new GlideRecord('incident');
rec.query();
while(rec.next()) {
    gs.print(rec.number + ' exists');
}
```
### update

```javascript
var rec = new GlideRecord('incident');
rec.addQuery('active', true);
rec.query();
while (rec.next()) {
    rec.active = false;
    gs.print('Active incident ' + rec.number + ' closed');
    rec.update();
}
```

### insert

```javascript
var rec = new GlideRecord('incident');
rec.initialize();
rec.short_description = 'Network problem';
rec.caller_id.setDisplayValue('Joe Employee');
rec.insert();
```

### delete

```javascript
var rec = new GlideRecord('incident');
rec.addQuery('active', false);
rec.query();
while (rec.next()) {
    gs.print('Inactive incident ' + rec.number + ' deleted');
    rec.deleteRecord();
}
```

### Querying Service Catalog Tables

You cannot directly query the variables of the Service Catalog Request Item table [sc_req_item]. Instead, query the Variable Ownership table, [sc_item_option_mtom], by adding two queries, one for the variable name and another for the value. The query returns the many-to-many relationship, which you can dot-walk to the requested item. The following example finds the request items that have the variable `item_name` with a value of `item_value` and displays the request item numbers:

```javascript
var gr = new GlideRecord('sc_item_option_mtom');
gr.addQuery('sc_item_option.item_option_new.name', 'item_name');
gr.addQuery('sc_item_option.value', 'item_value');
gr.query();
while (gr.next()) {
    gs.addInfoMessage(gr.request_item.number);
}
```

For additional information see `GlideRecord()` on the ServiceNow Developer Site.

### Running order guides automatically

Service catalog order guides allow customers to make a single service catalog request that can generate several ordered items.
Administrators can configure order guides to run automatically, from a workflow or a script to generate a set of ordered items without manually submitting a service catalog request. For example, an onboarding workflow for a new employee can run an order guide to automatically order items for that employee.

**Running order guides from scripts**

Running order guides with a server-side script is more complex than using workflows, but it allows more flexibility and can be used in non-workflow situations.

For example, you can use order guide scripts with UI actions or server-side business rules.

**Note:** When order guides run automatically, order guide UI policies are not enforced. Also, options in the Choose Options screen cannot be selected, so make sure that order guide rules define sensible defaults for these options to avoid processing failures.

Use the `SNC.ScriptableOrderGuide` Java class to run order guides with a script.

Use the `SNC.ScriptableOrderGuide(String orderGuideId)` constructor to create a new `ScriptableOrderGuide` object.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>process(String json)</code></td>
<td>boolean</td>
<td>Runs the order guide using the JSON encoded string parameter as the input for the order guide. Returns true or false depending on whether processing was successful or not.</td>
</tr>
</tbody>
</table>

**Note:** Both opened_by and requested_for parameters must be passed to the order guide, and both must have valid user record sys_id values.

If processing is successful and a request is created by the order guide, you can retrieve the request GlideRecord using `getRequest`.

If the processing fails, you can retrieve the failure GlideRecord using `getFailure()`, then submit the script for reprocessing using `reprocess`. 
<table>
<thead>
<tr>
<th>Method</th>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reprocess(GlideRecord failure)</td>
<td>boolean</td>
<td>Runs the order guide again using the JSON encoded string parameter stored in the failure GlideRecord.</td>
</tr>
<tr>
<td>getMessage()</td>
<td>String</td>
<td>Retrieves the message populated after processing or reprocessing.</td>
</tr>
<tr>
<td>getRequest()</td>
<td>GlideRecord</td>
<td>Retrieves the request GlideRecord.</td>
</tr>
<tr>
<td>getFailure()</td>
<td>GlideRecord</td>
<td>Retrieves the failure GlideRecord from the Scriptable Order Guide Failures [sc_script_order_guide_failure] table.</td>
</tr>
</tbody>
</table>

### Script Example

This script processes an order guide called IT Onboarding SOG.

```javascript
// Creating the object to later be JSON encoded
var json = {
"opened_by": "62826bf03710200044e0bfc8bcbe5df1", "requested_for": "06826bf03710200044e0bfc8bcbe5d8a", "department": "221f3db5c6112284009f4becd3039cc9"};

var gr = new GlideRecord("sc_cat_item_guide");
if (gr.get("name","IT Onboarding SOG")) {
    var sog = new SNC.ScriptableOrderGuide(gr.getValue("sys_id"));
    var result = sog.process(new JSON().encode(json));
    if(!result)
        gs.log("Processing the scriptable order guide failed with message: " + sog.getMessage());
    else {
        var request = sog.getRequest();
        gs.log("Request created - " + request.sys_id); } }
```

### Running order guides from workflows

Running an order guide from a workflow is suitable if you include order guides as part of a broader workflow-based process.

For example, an activity within an onboarding workflow for a new employee can automatically run an order guide to order items for that employee.

**Note:** When order guides run automatically, order guide UI policies are not enforced. Also, options in the Choose Options screen cannot be selected, so make sure that order guide rules define sensible defaults for these options to avoid processing failures.

To run order guides from a workflow, use the Scriptable Order Guideworkflow activity.
Table 1366: Running Order Guides from Workflows

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Guide</td>
<td>The name of the order guide that this activity processes. For example, Example Employee Onboarding IT.</td>
</tr>
<tr>
<td>Script</td>
<td>A script passing information to the order guide. This information is sent as a JSON encoded string parameter assigned to the answer variable.</td>
</tr>
</tbody>
</table>

The script must meet these requirements:
- The names of the variables in the script must match the names used within the order guide. For example, if the order guide uses a department variable in a rule condition, the script must also pass a department parameter.
- Both opened_by and requested_for parameters must be passed to the order guide, and both must have valid user record sys_id values.

Results

- Success: the activity successfully processed the order guide. This does not mean a request was created. If a request was created, the request sys_id is added to the workflow scratchpad under the sc_request variable.
- Failure: while processing the order guide a failure occurred, creating a failure record. If the processing fails, you can view and edit the failure record.

Workflow Example

The Example Employee Onboarding IT Workflow workflow uses this example to generate IT catalog items for a new employee as part of an onboarding process.

The activity uses this script to:
1. Take a JSON string generated previously from the HR change record.
2. Append the mandatory opened_by and requested_for parameters to that string.
3. Submit the new string for processing by the order guide.

```java
var parameters = new JSON().decode(current.payload);
// Need to amend the json object to include additional values.
parameters.opened_by = current.opened_by + "";
parameters.requested_for = current.opened_for + "";
answer = new JSON().encode(parameters);
```
Viewing order guide failures

Order guide processing may fail, for example if the order guide being run does not exist.

If a failure occurs, a failure record is created in the Scriptable Order Guide Failures table.

To view details of a failure, navigate to Service Catalog > Catalog Policy > Scriptable Order Guide Failures, then open a failure record.

Reprocessing Failures

If you have fixed the error that caused the initial failure, you can reprocess failed order guides.

2. Open the failure record.
3. Click the Reprocess related link.

To reprocess one or more failures:

2. Select the check box beside one or more records to reprocess.
3. Select Reprocess from the Actions choice list.

Scriptable assignment of execution plans

Each catalog item has an associated execution plan, used whenever an item of that type is ordered; if no plan is specified, the default plan is used. This default is effective for most organizations, but your execution plan may need to vary based on additional criteria.

For example, in the base system service catalog, a request for a new PC always uses the PC Delivery Plan. However, this plan may need to vary for unusual circumstances - such as when a requester is working from home, at a remote location.

To provide this flexibility, you can use a script to override the default execution plan on a specific catalog item.

Limitations during script execution

Execution plan scripts have limitations that need to be considered during their implementation.

While the execution plan script runs:

- You cannot interact with any catalog tasks as catalog tasks are only created after the execution plan is selected.
- Some fields such as total delivery time and due date are not yet calculated, although the request itself is available within the script via current.request().
- Approvals have not yet been generated.

Writing the scripts

Follow these guidelines when writing execution plan scripts.

Execution plan scripts can access the same global variables and other functions as in any other server side execution plan.

- current is the currently-requested catalog item, sc_req_item.
- current.delivery_plan() is the assigned execution plan for this catalog item.
The evaluated value from the script is used as the sys_id of the execution plan.

Simple example:

```javascript
current.delivery_plan.setDisplayValue('PC Delivery Plan')
```

If an invalid value is returned, such as undefined or not found, then the existing assigned value is used.

More complex example:

```javascript
getexecutionplan();
function getexecutionplan() {
  var location = current.request.requested_for.location.getDisplayValue();
  // if we're in Atlanta
  if (location == 'Atlanta') {
    // use the remote pc delivery plan instead of the normal one
    var remote_plan = new GlideRecord('sc_cat_item_delivery_plan');
    remote_plan.addQuery('name', 'Remote PC Delivery Plan');
    remote_plan.query();
    remote_plan.next();
    current.delivery_plan = remote_plan.sys_id;
    return remote_plan_sys_id;
  }
  return current_delivery_plan;
}
```

In this example, any time a request is for a user in Atlanta, ServiceNow uses the Remote PC Delivery Plan. Otherwise, the execution plan is not overridden and ServiceNow uses the catalog item's normal execution plan, the PC Delivery Plan.

### Add a script to a catalog item

You can add a script to a catalog item so that the script runs each time a user requests that item.

1. Navigate to **Service Catalog > Maintain Items**.
2. Select the relevant catalog item to which you wish to add the script.
3. Configure the catalog item form to add the execution plan script field, often named **Delivery Plan Script**.
4. Fill in the script details.
5. Update the item form with your changes.

The script runs each time that item is requested, selecting the execution plan to run with that item.
Scriptable service catalog variables

You can use scripting to reference any variables used by a request item from any table.

An example of a variable reference follows.

```javascript
current.variables.<variable_name>
```

Where `current` refers to the current record, and `<variable_name>` is the name of your variable.

**Note:** In order to reference a variable from JavaScript, it must have a name.

---

**Print a variable**

```javascript
var original = current.variables.original_number;
gs.print(original);
```

**Set a variable**

```javascript
current.variables.name = "Auto-Generated:" + current.variables.asset_tag;
```

**Create an inventory item with fields set from variables**

```javascript
doCreation();

function doCreation ( ) {
    var create = current.variables.create_item;
    if (create == 'true') { // we want to create an asset
        var computer = new GlideRecord('cmdb_ci_computer');
        computer.initialize();
        computer.asset_tag = current.variables.asset_tag;
        computer.serial_number = current.variables.serial_number;
        computer.name = current.variables.name;
        computer.manufacturer = current.variables.company;
        computer.insert();
    } }
```

---

**Notes and limitations**

1. You can only *set* a variable in a *before* business rule. Variables set in an *after* rule are not written to the database.

2. There is nothing in place to prevent namespace collision with variables. Creating two variables named `computer_speed` would result in only one of them showing up; the second one would overwrite the first one.

3. Date/time variables use the same time zone formatting and storage rules as all other dates in the system. They are stored internally in GMT, but translated into the user's local time zone and format for display.
Scripting currency and price fields

Administrators can use currency and price fields in scripts.

You can obtain currency values using:

- getReference methods: apply to the system's base currency, when the item was created for the service catalog.
- getCurrency methods: apply to the currency used by the customer who ordered the item.
- getSession methods: apply to the currency of the currently logged-in user viewing the item, for example, a member of a fulfillment group processing the order.

Working with values

Altering values

You cannot simply add an integer to the return values, as they are returned as string values.

For example, with a current.price of 708.32, current.price + 100 returns a value of 708.32100.

You can use a parseFloat function in your script to provide the correct result. For example, parseFloat(current.price) + 100 will return 808.32.

Returning predictable values

Adding values to prices with scripts can be unpredictable. For example, consider a script which adds 100 to the price of our item.

```javascript
var newPrice = parseFloat(current.price) + 100;
```

Running that script in the UK returns a value of var newPrice = parseFloat(708.32) + 100 = 808.32.

However, if you run it in the US, this returns a value of var newPrice = parseFloat(1000) + 100 = 1100.

To get the predictable values back into our item, regardless of the session under which the script runs, you can write your scripts to run in the system's base currency.

For example:

```javascript
var newPrice = parseFloat(current.price.getReferenceValue()) + 100;
current.price = current.price.getReferenceCurrencyCode() + ',' + newPrice;
```

**Note:** This example still relies upon knowing what the base currency is, for the 100 to be meaningful.

Obtaining currency values

The following methods allow you to obtain currency values.

The Example value returned column shows details for an item:

- Created with an item price of $600 in a system base currency of USD (US dollars)
- Ordered by a British user in the GBP (British pounds) currency
- Viewed and fulfilled by a Spanish user in the EUR (Euro) currency

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
<th>Example value returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>current.price.getReferenceCurrencyCode()</td>
<td>The currency ISO code, in the base system currency.</td>
<td>USD</td>
</tr>
<tr>
<td>current.price.getReferenceDisplayValue()</td>
<td>The price, including the currency symbol, in the base system currency.</td>
<td>$600.00</td>
</tr>
<tr>
<td>current.price.getReferenceValue()</td>
<td>A string representation of the field's value, in the system's basis currency.</td>
<td>600</td>
</tr>
<tr>
<td>current.price.getSessionCurrencyCode()</td>
<td>The currency ISO code, in the current user's currency.</td>
<td>EUR</td>
</tr>
<tr>
<td>current.price.getSessionDisplayValue()</td>
<td>A string representation of the field's value, including the currency symbol, in the current user's currency.</td>
<td>€449,75</td>
</tr>
<tr>
<td>current.price.getSessionValue()</td>
<td>A string representation of the field's value, in the current user's currency.</td>
<td>449,75</td>
</tr>
<tr>
<td>current.price.getCurrencyCode()</td>
<td>The currency ISO code, in the currency used when the item was ordered.</td>
<td>GBP</td>
</tr>
<tr>
<td>current.price.getCurrencyDisplayValue()</td>
<td>A string representation of the field's value, including the currency symbol, in the currency used when the item was ordered.</td>
<td>£373.83</td>
</tr>
<tr>
<td>current.price.getCurrencyString()</td>
<td>A string including the ISO code as well as the price, in the currency used when the item was ordered.</td>
<td>GBP;373.83</td>
</tr>
<tr>
<td>current.price.getCurrencyValue()</td>
<td>The price, in the currency used when the item was ordered.</td>
<td>373.83</td>
</tr>
</tbody>
</table>

**Note:** Currency symbols are displayed on the left or the right of the currency as appropriate.

**Examples**

```
current.price=1000
```
This sets the current price = 1,000 of whatever the current session currency is. For example, for a Japanese user, the price is JPY (Japanese Yen) 1,000.

```
current.price='USD;1000'
```

Sets the current price = 1,000 USD.

**Service catalog script API**

A scriptable API for the service catalog makes it easier to order from the catalog when using business rules.

The `Cart()` API allows you to order any quantity of catalog items, using the `sys_id` of the Catalog Item `[sc_cat_item]` you want. You can then set catalog variables to values as required, assuming the variables have names.

**Note:** If the script runs as a result of a scheduled import, the script runs as system or as the user specified by that import in the **Run as** field. The script uses the specified user’s cart. Each instantiation of a new `Cart()` object empties the cart of the calling user, but does not use or empty any other carts.

**Examples**

Ordering a single BlackBerry:

```
var cartId = GlideGuid.generate(null);
var cart = new Cart(cartId);
var item = cart.addItem('e2132865c0a8016500108d9cee411699');
var rc = cart.placeOrder();
gs.addInfoMessage(rc.number);
```

Ordering twelve BlackBerries:

```
var cartId = GlideGuid.generate(null);
var cart = new Cart(cartId);
var item = cart.addItem('e2132865c0a8016500108d9cee411699',12);
var rc = cart.placeOrder();
gs.addInfoMessage(rc.number);
```

Ordering an executive desktop and setting its OS:

```
var cartId = GlideGuid.generate(null);
var cart = new Cart(cartId);
var item = cart.addItem('e46305bdc0a8010a00645e608031eb0f');
cart.setVariable(item,'os','Linux Red Hat');
var rc = cart.placeOrder();
gs.addInfoMessage(rc.number);
```

**Setting a GlideRecord variable to null**

GlideRecord variables (including current) are initially null in the database. Setting these back to an empty string, a space, or the JavaScript null value will not result in a return to this initial state.

**Note:** This functionality requires a knowledge of JavaScript.
Note: Functionality described here requires the Admin role.

To set it back to the initial state, simply set the value to "NULL". Note that the update() function does not run on the current object but rather on the record. The object displays the initial value until it is called again from the record.

Example 1

```javascript
var gr1 = new GlideRecord('incident');
gr1.query();
while(gr1.next()) {
  gr1.priority = "NULL";
  gr1.update();
}
```

Example 2 (Business Rule)

```javascript
current.u_affected_value = 'NULL';
current.update();
```

Use a script to approve an execution plan

You can use an approval rule script to approve an execution plan.

1. Retrieve an approval execution plan task.
2. View the Approval Script field.
3. Fill in an approval script using the same syntax and rules you would use on an approval rule.

For example, in the script below, the requestor's manager is the approver.

```
var request = current.request_item.request;
var rc = request.requested_for.manager;
rc;
```
Wizard scripts

You can create scripts to implement advanced custom functionality using wizards.

Use wizard scripts in record generator panels, transitions, and UI policies.

**Note:** The System Wizards application is not active by default.

To reference a wizard variable in a script, use the format `wizard.<name>`, where `<name>` is the value in the Name field of the variable definition.

Wizard script examples

These examples demonstrate scripts that use wizard variables.

**Example 1**

```javascript
//Find a user in the sys_user table with the wizard variable 'EmpName'
function getUser() {
    var gu = new GlideRecord('sys_user');
    gu.addQuery('sys_id', wizard.EmpName);
    gu.query();
    if (gu.next()) {
        gs.addInfoMessage('Found user ' + gu.name);
    }
}
```

**Example 2**

```javascript
//Loop through items in a wizard list collector with variable name astList
function removeAssets() {
    var items = wizard.astList.toString();
    items = items.split(',');
    for (var i = 0; i < items.length; i++) {
        // got the asset id
        var sys_id = items[i];
    }
}
```

**Example 3**

```javascript
//Set the url on wizard completion - user will be redirected to this location
var uri = ('sys_user_list.do?sysparm_query=sys_id=' + wizard.EmpName);
wizard.redirect = uri;
```

Wizard scripts on record generators

Use wizard scripts in record generator panels to create records in any table, including on more than one table.

1. In the **Table** field, select **Global**.
2. Enter a script in the **Script** field. Configure the form to add the field, if necessary.
Client scripts

Client scripts run on the client (web browser). You can use client scripts to define custom behaviors that run when events occur such as when a form is loaded or submitted, or a cell changes value.

Proper client-side processing depends on the form loading completely first. Making record updates prior to form load can produce unexpected results that bypass client-side processing.

Client-side Glide APIs (Application Programming Interfaces) provide classes and methods that you can use in scripts to perform client-side tasks.

Client script basics

Review the basics of ServiceNow platform client scripting.
Client script types on page 3901
Client script access to variable fields on task records on page 3902
Create a catalog client script on page 3906
GlideAjax on page 3926
Mobile Client GlideForm (g form) scripting on page 3922
Introducing Client Scripting (Video)

UI scripting

Use scripts to create custom UIs.

- UI scripts on page 3913
- Global UI scripts on page 3914
- Wizard UI policy and client scripts on page 3915
- Displaying a custom dialog on page 3908

Jelly scripting

Use Jelly in ServiceNow platform scripts.

- Extensions to Jelly syntax on page 3937
- Scripting of field types on page 3810
- Jelly escaping types on page 3944
- Introducing Jelly Scripting - Part 1 (Video)
- Introducing Jelly Scripting - Part 2 (Video)
- Introducing Jelly Scripting - Part 3 (Video)

Client script types

Client scripts are shipped to the client (the browser) and run there instead of on the server.

Users who know JavaScript can define scripts to run in the client browser. Several types of scripts are supported:

- **onLoad()**: Runs when a form is loaded.
- **onChange()**: Runs when a particular widget changes value.
- **onSubmit()**: Runs when a form is submitted.
- **onCellEdit()**: Runs when a cell on a list changes value.

**Note:** Best practice for performance reasons is to limit the amount of code sent to the client. You should always be aware of the impact on the client when adding client scripts.

**Warning:**

- Warning:
  - With the exception of onCellEdit client scripts, UI policies and client scripts apply to forms only. If you create UI policies or client scripts for fields on a form, you must use another method to ensure that data in those fields is similarly controlled in a list. You can:
    - Disable list editing for the table.
• Create appropriate business rules or access controls for list editing.
• Create data policies.
• Create a separate onCellEdit client script.

Client script access to variable fields on task records

Variable fields may be accessed via client script on records that are based on the task record.

The most obvious use for this is making Service Catalog variables conditionally display and function in the same way as they did when initially creating an order when they subsequently appear on Request [sc_request], Requested Item [sc_req_item], and Catalog Task [sc_task] records.

The following API is supported via g_form:

- g_form.setDisplay(name, display)
- g_form.setVisible(name, visibility)
- g_form.setMandatory(name, mandatory)
- g_form.setValue(name, value, display_value)
- g_form.getValue(name)
- g_form.setReadOnly(fieldName, boolean)

To specify that a variable is to be the object of the API call, the name must be prefixed with "variables.". This is necessary as another field that is not a variable but has the same name may already be present on the form.

For example, if we ordered an Executive Desktop using the out-of-box Service Catalog, we could add a client script to the Request Item [sc_req_item] to disable the display of the "cpu_speed" variable one would create a client script containing:

```
g_form.setDisplay("variables.cpu_speed", false);
```

Hiding mandatory variables

To hide a mandatory variable using a script, the script must first set the mandatory property of the variable to false.

Client script FAQ

Frequently asked questions and answers regarding client scripting.

Why do I have to name scripts like that?

The client code needs to know which function to attach to the document. Hence, it requires that you follow our naming convention with functions. Note that you can have more than one function on a client side script and have your onSubmit() function, for example, call other functions. The requirement is that the primary function follow our naming convention.

Can I have more than one function listening to the same thing?

You can, but there is no guarantee of sequencing. You cannot predict what order your event handlers will run.

How do I reference a field on the form?

To reference the actual HTML element of the priority field on the incident form:

```
var priority_field = g_form.getControl('priority');
```
The easiest way to reference a field's value is to use the built-in `g_form()` helper variable:

```javascript
var value_of_priority = g_form.getValue('priority');
```

**How do I cancel a form submission?**

Have your `onSubmit()` function return `false`. For example the following snippet of code disallows all form submission.

```javascript
function onSubmit() { return false; }
```

**How do I confirm a field value with the user?**

```javascript
function onSubmit() { //This allows you to access fields from the referenced record
    var assignedTo = g_form.getReference('assigned_to');
    var answer = confirm("The Assigned to is " + assignedTo.user_name + ". Is this correct?");
    if(answer == true) { return false; }
}
```

**How do I force a Save and Stay?**

```javascript
// onChange event on Incident table using Field name "Caller"
function onChange(control, oldValue, newValue, isLoading) {
    // Setting the control value prevents against an infinite loop
    // Change caller_id to the field the change event reacts to
    if(control.caller_id != newValue && control.caller_id != null) {
        //do any comparisons you want
        // . . . //
        //does an update and stay
        gsftSubmit(document.getElementById('sysverb_update_and_stay')); }

    // Change caller_id to the field the change event reacts to
    control.caller_id = newValue; }
```

**How do I call a business rule from a client script?**

To call a business rule from a client script, use `GlideAjax()`.

**Client script form**

Field descriptions for the client script form.

To open the client script form, navigate to **System Definition > Client Scripts**.

You can add client scripts only on tables that are in the same scope as the client script (starting with the Fuji release).
Figure 938: Client script form

The client script form provides the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unique, descriptive name for this script.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application</td>
<td>Application that contains this client script (starting with the Fuji release).</td>
</tr>
<tr>
<td>Active</td>
<td>Indicator of whether the script is active.</td>
</tr>
<tr>
<td>Global</td>
<td>Indicator of whether the script applies to all views. When <strong>Global</strong> is selected, the script runs regardless of what view is being displayed. When <strong>Global</strong> is not selected, this script runs only if the current view matches the script's view.</td>
</tr>
<tr>
<td>View</td>
<td>View to which the script applies. This option is available only if <strong>Global</strong> is not selected.</td>
</tr>
</tbody>
</table>
| Type         | Indicator of when the script runs. Select one of the following types:  
  • **onLoad()**: when a form is first loaded.  
  • **onSubmit()**: when a form is submitted. This type allows you to cancel the submission, if necessary.  
  • **onChange()**: when a specified field on the form changes value.  
  • **onCellEdit()**: when a user interacts with a specified field in a list.                                                                 |
| Table        | Table or database view to which this script applies, such as incident or change_request.                                                                                                                |
| Inherited    | Indicator of whether this script applies to any extended tables. For example, if the script is active on the task table, selecting **Inherited** runs this script on the incident, change_request, and problem tables, as well. |
| Field name   | Field to which the script applies. This option is available only if **Type** is set to **onChange()**.                                                                                                         |
| Description  | Detailed description of the client script.                                                                                                                                                                |

**Note:** The list shows only tables and database views that are in the same scope as the client script (starting with the Fuji release).
Create a catalog client script

Follow this procedure to create a catalog client script.

1. Navigate to Service Catalog > Catalog Policy > Catalog Client Scripts. A list of current custom catalog client scripts appears.
2. Click New.
3. Fill in the fields, as appropriate (see table).
4. Click Submit.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a unique name for the catalog client script.</td>
</tr>
<tr>
<td>Applies to</td>
<td>Select the item type this client script applies to:</td>
</tr>
<tr>
<td></td>
<td>• A Catalog Item: enables the Catalog item field.</td>
</tr>
<tr>
<td></td>
<td>• A Variable Set: enables the Variable set field.</td>
</tr>
<tr>
<td>Active</td>
<td>Select the check box to enable the client script. Clear the check box to</td>
</tr>
<tr>
<td></td>
<td>disable the script.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UI Type</td>
<td>Whether to apply this to desktop, mobile, or both.</td>
</tr>
<tr>
<td>Script</td>
<td>Enter the client script that should run on the service catalog item.</td>
</tr>
<tr>
<td>Type</td>
<td>Select when the script should run, such as <strong>onLoad</strong> or <strong>onSubmit</strong>.</td>
</tr>
<tr>
<td>Catalog item or Variable set</td>
<td>Select a catalog item or variable set from the list. The field name and options available depend on the selection in the <strong>Applies to</strong> field.</td>
</tr>
<tr>
<td>Applies on a Catalog Item view</td>
<td>Select the check box to apply the catalog client script to catalog items displayed within the order screen on the service catalog.</td>
</tr>
<tr>
<td>Applies on Requested Items</td>
<td>Select the check box to apply the catalog client script on a Requested Item form, after the item is requested.</td>
</tr>
<tr>
<td>Applies on Catalog Tasks</td>
<td>Select the check box to apply the catalog client script when a Catalog Task form for the item is being displayed.</td>
</tr>
</tbody>
</table>

**Catalog client script creation**

Client-side scripts can add dynamic effects and validation to forms. Scripts can apply to service catalog items or variable sets, allowing administrators to use the same functionality that is available on other forms.

You can use client side scripts to:

- Get or set variable values.
- Hide or display variables.
- Make variables mandatory or not.
- Validate form submission.
- Add something to the cart.
- Order something immediately.

**Catalog client script considerations**

When you create catalog client scripts, be aware of the following considerations.

- Catalog client scripts run when a user orders an item from the service catalog. Catalog client scripts can also run when variables or variable sets for a catalog item are displayed when a user requests that item.
- For a variable to be accessible using a catalog client script, it must have a variable name. Variables without names do not appear in the list of available variables.
- When using standard client scripts on a Requested Item or Catalog Task form, make a note of fields with the same name as variables. If a table field and a variable of the same name are both present on a form, the table field is matched when it is accessed using a script. If this happens, specifically address the variable by naming it `variables.variable name`. For example:
  ```javascript
  g_form.setValue('variables.replacement', 'false');
  ```
- If you are using record producers to pass variables from the service catalog to other types of records, these variables are made visible in those records with a variable editor, such as the Change Variable Editor UI formatter on Change request forms. You can manipulate these variables using standard client script methods, such as `setDisplay`, `setMandatory`, `setValue`, and `getValue`.
- Catalog client scripts can be used for catalog items included in a wizard.
Catalog client script differences
Catalog client scripts are very similar to standard client scripts, with a few important differences.

<table>
<thead>
<tr>
<th>Instead of selecting a table such as Incident for the script, select a catalog item or variable set. As your system may have a large number of catalog items, you should select a catalog item or variable set using a reference field instead of the choice list that the standard Client Script form uses.</th>
</tr>
</thead>
<tbody>
<tr>
<td>When using an onChange() catalog client script, it is linked to a particular variable instead of a field. The system automatically populates the variable name selection list with any named variables from the catalog item or variable set selected.</td>
</tr>
</tbody>
</table>

Catalog client script examples
Examples of client scripts to perform common actions.

**Get the value of a variable**
Use the following syntax to obtain the value of a catalog variable. Note that the variable must have a name. Replace `variable_name` with the name of the variable.

```javascript
g_form.getValue('variable_name');
```

**Restrict the number of characters a user can type in a variable**
This is an example of a script that runs when the variable is displayed, rather than when the item is ordered.

```javascript
function onLoad() {
  var sd = g_form.getControl('short_description');
  sd.maxLength=80;
}
```

Displaying a custom dialog
Using `GlideDialogWindow()`, you can write a script to open a custom dialog.

A GlideDialogWindow can be displayed anywhere you use client scripts, UI macros, or UI actions. These are most commonly called from a UI action with the `Client` check box selected.

For reference information on the available `GlideDialogWindow()` methods, see *GlideDialogWindow()* API reference.

**GlideDialogWindow basics**

Your script should accomplish the following tasks to display a dialog window:

- Initialize the window by calling `GlideDialogWindow(id)`.
- Specify the table to display in the dialog with `setPreference(name, value)`.
- Display the dialog using `render()`.

You can call `setTitle(title)` and `setSize(w,h)` to specify the window title and size. If you do not call `setTitle(title)`, the dialog is created without a title displayed in the title bar. You can also define a UI page to display in the dialog window and pass the result back to the calling window.
Displaying a list of records

This example opens a list of incidents in a dialog window. When the user selects an item in the dialog window, that item is opened in the calling window or, if the user clicks the close box, the window closes.

To trigger the dialog window, define a UI action or UI macro with a script that initializes the dialog, references the UI page, and renders the dialog window. In this example, a UI action to define and trigger the dialog window was created with the following attributes and script:

- **Name**: Popup Display List
- **Table**: Incident
- **Action name**: popup_display_list
- **Form button**: True
- **Client**: True
- **Onclick**: popupDispList()

```javascript
function popupDispList() {
    // Initialize the GlideDialogWindow
    var gdw = new GlideDialogWindow('display_incident_list');
    gdw.setTitle('Incidents');
    gdw.setPreference('table', 'incident_list');
    gdw.setPreference('sysparm_view', 'default');

    // Set the table to display
    var num = g_form.getValue('number');
    var query = 'active=true^priority=1^number!=' + num;
    gdw.setPreference('sysparm_query', query);

    // Open the dialog window
    gdw.render();
}
```

Opening a custom UI page in a dialog window

This is an example of how to use GlideDialogWindow() to open a custom UI page in a dialog window and pass the results back to the calling window.

To accomplish this, you will need to create the following:

- A trigger to open the dialog window and reference the UI page.
- A UI page with HTML and client script definitions.
- A script include to escape reserved characters in XML.

Defining the trigger

To trigger the dialog window, define a UI action or UI macro with a script that initializes the dialog, references the UI page, and renders the dialog window.

In this example, the following UI action triggers the dialog window:

- **Name**: Comments Dialog
- **Table**: Incident
- **Action name**: comments_dialog
function commentsDialog() {
    // Get the values to pass into the dialog
    var comments_text = g_form.getValue("comments");
    var short_text = g_form.getValue("short_description");

    // Initialize and open the dialog
    var dialog = new GlideDialogWindow("add_comments_dialog"); // Instantiate the dialog containing the UI page 'add_comments_dialog'
    dialog.setTitle("Add Task Comments"); // Set the dialog title
    dialog.setPreference("comments_text", comments_text); // Pass the comments into the dialog
    dialog.setPreference("short_text", short_text); // Pass in a short description for use in the dialog
    dialog.render(); // Open the dialog
}

The script obtains the values to pass to the dialog, calls the JSUtil script include to escape any content, initializes the dialog attributes, and displays the dialog.

**Defining the UI page**

In addition to the trigger that opens the dialog window, you will need to define the UI page to display in the dialog window. The UI page name must match the GlideDialogWindow that is instantiated in the trigger script. The example triggering script in the previous section creates a dialog window "add_comments_dialog" so the UI page is named "add_comments_dialog".

The following example UI page is displayed in the dialog window to add comments to an incident. See [Extensions to Jelly Syntax](#) for information on the ServiceNow extensions to Jelly. To obtain the window properties from the triggering script defined by the setPreference() method into the UI page, use RP.getWindowProperties() in the HTML script.

**Note:** For scoped applications, the RP.getWindowProperties.get() method is not available. Instead use the syntax RP.getWindowProperties.<property>, where <property> is the dialog box property that you want to obtain. For example, RP.getWindowProperties().get('title') would be RP.getWindowProperties().title.

**HTML script**

```html
<g:ui_form>
    <!-- Get the values from dialog preferences -->
    <g:evaluate var="jvar_short_text"
        expression="RP.getWindowProperties().get('short_text')" />
    <g:evaluate var="jvar_comments_text"
        expression="RP.getWindowProperties().get('comments_text')" />
    <!-- Set up form fields and labels -->
    <table width="100%"
        <tr id="description_row" valign="top">
            <td colspan="2">
                <!-- Short description value used as a label -->
            </td>
        </tr>
```

© 2017 ServiceNow. All rights reserved. 3910
Client script

The following is an example of a client script that is called when the OK button is clicked in the dialog window. The script checks if there is data in the dialog and, if so, closes the dialog window and pass the comments entered in the dialog window into the Comments field of the original form.

```javascript
function validateComments() {
  // This script is called when the user clicks "OK" in the dialog window
  // Make sure there are comments to submit
  var comments = gel("dialog_comments").value;
  var ga = new GlideAjax('validateComments'); // Call script include to escape text
  ga.addParam('sysparm_name', 'validateComments');
  ga.addParam('sysparm_comments', comments);
  ga/XMLWait();
  comments = ga.getAnswer(); // Set comments to escaped text
  comments = trim(comments);
  if (comments == "") {
    // If comments are empty, alert the user and stop submission
    alert("Please enter your comments before submitting.");
    return false;
  }
  // If there are comments, close the dialog window and submit them
  GlideDialogWindow.get().destroy(); // Close the dialog window
  g_form.setValue("comments", comments); // Set the "Comments" field with comments in the dialog
}
```
Script Include

To escape out XML-reserved characters from the content users enter into the custom dialog, create this script include.

- **Name**: validateComments
- **Client callable**: True
- **Script**:

```javascript
var validateComments = Class.create();
validateComments.prototype = Object.extendObject(AbstractAjaxProcessor, { 
    validateComments: function() { 
        var comments = this.getParameter('sysparm_comments');
        var ampRegex = new SNC.Regex('/&/');
        var ltRegex = new SNC.Regex('/</');
        var gtRegex = new SNC.Regex('/>/');

        var result = ampRegex.replaceAll('' + comments, '&' + 'amp;');
        result = ltRegex.replaceAll(result, '&' + 'lt;');
        return gtRegex.replaceAll(result, '&' + 'gt;');
    },
    type: 'validateComments'
});
```

Session client data

Session client data is a set of named strings sent from the server to the client when a form loads. Client scripts can store this data gathered from the server during form load, which is faster than an AJAX call to the server when the data is needed.

**Note**: To use session client data, the form must be configured to add the Synchronous field, which requires special setup by a TSE to enable the necessary synchronous event processing.

Using session client data

A script can use session client data to determine a user’s groups and group roles while the session is established. Session client data provides this information to the client script without requiring an AJAX call, thereby speeding up form load time.

When a session is established, a `session.established` event is added to the event queue. A script action that responds to this event can be written to set up any necessary client data for the session, as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Suggested Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Set session client values</td>
</tr>
<tr>
<td>Event name</td>
<td><code>session.established</code></td>
</tr>
<tr>
<td>Active</td>
<td>Checked</td>
</tr>
</tbody>
</table>
The GlideSession API includes:

```javascript
putClientData(name, value);
getClientData(name);
clearClientData(name);
```

When a form is created on the server, the Session Client Values are read and added to the `g_user` object that is created on the client. From any client script, you can use this JavaScript to get the client session data:

```javascript
var v = g_user.getClientData('test1');
```

In this case, the variable `v` has a value of `value1`.

For examples from a ServiceNow product principal on how you can reduce client-side lookups, see the Two ways to Reduce Client Side lookups blog post in the ServiceNow Community.

UI scripts

UI scripts provide a way to package client-side JavaScript into a reusable form, similar to how script includes store server-side JavaScript. Administrators can create UI scripts and run them from client scripts and other client-side script objects and from HTML code.

Create a UI script

Create a UI script to define reusable client-side JavaScript code.

To create UI scripts, navigate to System UI > UI Scripts and create or edit a record (see table for field descriptions).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Script Name</td>
<td>Name of the UI script. Ensure the name is unique on your system.</td>
</tr>
<tr>
<td>API Name</td>
<td>The API name of the UI script, including the scope and script name (for example, x_custom_app.HelloWorld).</td>
</tr>
<tr>
<td>Application</td>
<td>Application that contains the UI script.</td>
</tr>
<tr>
<td>Active</td>
<td>Indicator of whether the UI script is active. Only active UI scripts can run.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Global</td>
<td>Indicator of whether the script loads on every page in the system.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Use caution when creating global UI scripts because they can impact performance. You cannot create a global UI script in a scoped application.</td>
</tr>
<tr>
<td>Description</td>
<td>Summary of the purpose of the script.</td>
</tr>
<tr>
<td>Script</td>
<td>Client-side script to run when called from other scripts.</td>
</tr>
</tbody>
</table>

**Run UI scripts**
Follow these guidelines when running UI scripts.

To run a UI script on a form, create a formatter. In the associated UI macro, include a `g:requires` tag and specify the `name=` parameter as the name of the UI script followed by the `.jsdbx` extension. Add the formatter on the form view.

This code ensures that the definitions and results of the UI script are immediately available in the browser.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null"
xmlns:g2="null">
    <g2:evaluate var="jvar_stamp">
        var gr = new GlideRecord('sys_ui_script');
        gr.orderByDesc('sys_updated_on');
        gr.query();
        gr.next();
        gr.getValue('sys_updated_on');
    </g2:evaluate>
    <g:requires name="<UI SCRIPT NAME>.jsdbx" params="cache="$[jvar_stamp]" / >
</j:jelly>
```

To run a UI script from HTML code, use the `<script>` tag and specify the `src=` argument as the name of the UI script followed by the `.jsdbx` extension. For example, include the UI script named CoolClock with this code:

```html
<script language="javascript" src="CoolClock.jsdbx" /></script>
```

**Global UI scripts**
You can create a UI script and designate it as global, which makes the script available on any form in the system. You cannot create a global UI script in a scoped application.

You can mark a UI script as Global to make it available on any form in the system. You cannot create a global UI script in a scoped application. For example, you can create a UI script that has a function `helloWorld()`, and has the Global field checked:

```javascript
function helloWorld() {
    alert('Hi');
}
```
After you create this global UI script, you can call the `helloWorld()` function from any client script or UI policy you write.

**Wizard UI policy and client scripts**

Administrators and users with appropriate access rights can create dynamic effects for wizards using UI policies and client scripts, including:

- Get or set variable values
- Hide or display variables
- Make variables mandatory
- Validate form submission

UI policies apply effects based on conditions constructed with a condition builder. Client scripts accomplish more advanced functionality. Because UI policies do not require scripting, they are less likely to need maintenance after system updates.

**Note:** The System Wizards application is not active by default.

**Wizard UI policy**

UI policies can be applied to wizards.

1. Navigate to System Wizards > Wizard Policy > Wizard UI Policy.
2. Click New or select the policy to edit.
3. Enter the UI policy details and save the record.

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wizard</td>
<td>Select the wizard to which the UI Policy applies</td>
</tr>
<tr>
<td>Reverse if false</td>
<td>Select the check box to reverse the UI policy if the wizard condition statement evaluates to false.</td>
</tr>
<tr>
<td>Order</td>
<td>Enter the sequence in which this condition is evaluated if more than one matching condition exists. The order is evaluated from the lowest value to the highest value.</td>
</tr>
<tr>
<td>Global</td>
<td>This field is not used for wizards.</td>
</tr>
<tr>
<td>Short description</td>
<td>Enter a brief description.</td>
</tr>
<tr>
<td>Wizard Conditions</td>
<td>Create conditions for the UI policy using wizard variables. The policy is applied if the conditions evaluate to true.</td>
</tr>
<tr>
<td>On load</td>
<td>Select the check box to apply the UI policy when the form is loaded. Clear the check box to apply the policy only when the form is changed.</td>
</tr>
</tbody>
</table>
Field | Input Value
--- | ---
Run scripts | Select the check box to use the Execute if true and Execute if false scripting fields. Scripts are necessary to apply a UI policy other than Read Only, Mandatory, or Visible. For example, you must create a script to apply a UI policy for a specific role.
Active | Select the check box to enable the UI policy. Clear the check box to disable it.
Inherit | Select the check box for other tables to inherit the UI policy. This option only applies to custom tables that extend the expert_ui_policy table.

**Wizard client scripts**
Wizard client scripts can create advanced dynamic effects.

The following table summarizes differences between standard and wizard client scripts.

<table>
<thead>
<tr>
<th>Standard client scripts</th>
<th>Wizard client scripts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply to specific table</td>
<td>Apply to specific wizard and panel</td>
</tr>
<tr>
<td>Apply to specific field in the table</td>
<td>Apply to specific variable in the wizard</td>
</tr>
</tbody>
</table>

**Note:** Variables must have a name to be accessible in client scripts.

1. Navigate to **System Wizards > Wizard Policy > Wizard Client Scripts**.
2. Click **New** or select the client script to edit.
3. Enter the client script details and save the record.

![Wizard client script](image)

**Figure 939: Wizard client script**
Examples

To get the value of a variable:

```javascript
g_form.getValue('variable_name');
```

To restrict the number of characters a user can enter in a variable:

```javascript
function onLoad() {
    var sd = g_form.getControl('variable_name');
    sd.maxLength = 80; }
```

**Note:** Because wizard panels are different from forms, not all `g_form()` methods function properly in wizard client scripts. Test all methods to determine whether they function properly before using them.

What are some of the available helper functions?

Available helper functions.

**What are some of the available helper functions?**

There is a global object that ServiceNow automatically creates called `g_form` that references the currently active form. You can use the `g_form` object in your scripts to simplify certain common programming tasks.

A second global object, called `g_user`, references the currently active user. Just like `g_form`, you can use `g_user` in your scripts to react based on the current user (and that user's roles).

**Hide or show a field**

While you can use client scripting to hide or show a field, you can usually accomplish the same thing without scripting via UI policies.

```javascript
g_form.setVisible(<fieldname>, <value>);
```

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    var index = newValue.indexOf('Change');
    if(index == -1 ) { // hide it
        g_form.setVisible('field', false); }
    else { // unhide it
        g_form.setVisible('field', true); } }
```

**Make a field mandatory**

While you can use client scripting to make a field mandatory, you can usually accomplish the same thing without scripting via UI policies.

```javascript
g_form.setMandatory(<fieldname>, <value>);
```

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    if(oldValue == newValue) return;
    g_form.setMandatory('short_description', true); }
```

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    if(isLoading) return;
    if(newValue == "Critical")
        g_form.setMandatory('u_due_date', true);
    else
```
Make a field read-only

While you can use client scripting to make a field read-only, you can usually accomplish the same thing without scripting by using UI policies.

In order to make a field read-only, the field must be present on the form, though it may be hidden. The following example makes the `incident_state` field read-only on the Incident form upon load:

```javascript
function onLoad() {
    g_form.setReadonly('incident_state', true); }
```

To do the same thing for a derived field (e.g., the Caller's phone number field if it has been added to the Incident form), you can use the following:

```javascript
function onLoad() {
    g_form.setReadonly('caller_id.phone', true) }
```

Get a field value

```javascript
g_form.getValue(<fieldName>);
```

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    alert(g_form.getValue('short_description')));
}
```

Set a field value

```javascript
g_form.setValue(<fieldName>, <value>);
```

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    g_form.setValue('short_description', 'hello world'); }
```

Set a glide_list field value

You can pass one argument or two. For reference values, passing both the sys_id and the display value is more efficient because the client does not have to look up the display values for you. However, if you do not want to pass the display value, the client will do an AJAX call to find the value, which takes time.

Example: reset to no values.

```javascript
g_form.setValue("watch_list", "");
```

Example: Set to two references, passing the display values to avoid an AJAX call.

```javascript
var valueArray = new Array("46d44a23a9fe19810012d100cca80666", "46c6f9efa9fe198101dd5eed9aff6e7");
var labelArray = new Array("Beth Anglin", "Bud Richman");
g_form.setValue("watch_list", valueArray, labelArray);
```

Example: Set arbitrary email address (for a sys_user glide list).

```javascript
g_form.setValue("watch_list", ",", "you@you.com");
```

Example: Set references and emails, passing the display values to avoid an AJAX call.

```javascript
var valueArray = new Array("46d44a23a9fe19810012d100cca80666", ",", "46c6f9efa9fe198101dd5eed9aff6e7");
var labelArray = new Array("Beth Anglin", "me@me.com", "Bud Richman");
```
g_form.setValue("watch_list", valueArray, labelArray);

Example: Set references and emails, but use AJAX to obtain display values for references (this is a bit less efficient as the server must be contacted for the display values).

var valueArray = new Array("46d44a23a9fe19810012d100cca80666", "me@me.com", "46c6f9efa9fe198101ddf5eed9ad6e7");
g_form.setValue("watch_list", valueArray);

Example: Set references and emails using a comma separated string, but use AJAX to obtain display values for references (this is a bit less efficient as the server must be contacted for the display values).

g_form.setValue("watch_list", "46d44a23a9fe19810012d100cca80666,me@me.com," 46c6f9efa9fe198101ddf5eed9ad6e7");

Add an Option to the end of a choice list

`g_form.addOption(<fieldName>, <choiceValue>, <choiceLabel>);`

```javascript
function onChange(control, oldValue, newValue, isLoading) {
  g_form.addOption('priority', '6', '6 - Really Low'); }
```

Add an Option at a specific point on a choice list

`g_form.addOption(<fieldName>, <choiceValue>, <choiceLabel>, <targetIndex>);`

```javascript
function onChange(control, oldValue, newValue, isLoading) {
  g_form.addOption('priority', '2.5', '2.5 - Really Low', 3); }
```

Note that an insert happens at the location you specify on a zero based array. Existing options will all shift up one level. For example:

[A, B, C, D]
Insert E at 0
[E, A, B, C, D]
Insert E at 2
[A, B, E, C, D]

Remove an option from a choice list

```javascript
function onChange(control, oldValue, newValue, isLoading) {
  g_form.addOption('priority', '2.5', '2.5 - Really Low', 3); }
```

```javascript
function onChange(control, oldValue, newValue, isLoading) {
  g_form.removeOption('priority', '1'); }
```

Remove the Closed state option if the user is not an admin:

```javascript
function onLoad() {
  var isAdmin = g_user.hasRole('admin');
  var state = g_form.getValue('state');
  if(!isAdmin && (state != 7)) {
    alert('Current user is not an admin');
    g_form.removeOption('state', '7'); }
}
```

Remove ALL options from a choice list
The **Category** choice list, for example:

```javascript
var choiceValue = g_form.getValue(<fieldName>);
var choiceLabel = g_form.getOption(<fieldName>, choiceValue).text;
```

**Get the label for a choice list value**

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    var choiceValue = g_form.getValue('category');
    var choiceLabel = g_form.getOption('category', choiceValue).text;
}
```

Show or hide an action button depending on the form view

```javascript
function onLoad() {
    var objView = document.getElementById('sysparm_view');
    var strView = objView.getAttribute('value');
    if(strView == 'ess') {
        var objCloseButton = document.getElementById('close_incident');
        objCloseButton.style.display = 'none';
    }
}
```

**Look up the value of a reference field**

```javascript
var gr = g_form.getReference('fieldName');
```

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    var caller = g_form.getReference('caller_id');
    if(caller.vip == 'true')
        alert('Caller is a VIP!');
}
```

**Note:** When setting values, use the storage value instead of the display value.

**Get the current user name**

```javascript
var myVariable = g_user.userName;
```

```javascript
function onLoad() {
    var userName = g_user.userName;
    alert('Current user = ' + userName);
}
```

**Get the value of a parameter**

```javascript
var currentView = g_form.getParameter('view');
```

```javascript
function onLoad() {
    var view = g_form.getParameter('view');
    alert('Current view = ' + view);
}
```

**Test if the current user has a role**

```javascript
var myRole = g_user.hasRole(<role>);
```

```javascript
function onLoad() {
    var isAdmin = g_user.hasRole('admin');
    if(isAdmin)
        alert('Current user is an admin');
}
else
  alert('Current user is NOT an admin'); }

Determine what submit button was clicked

function onSubmit() {
  var action = g_form.getActionName();
  alert('You pressed ' + action); }

Determine if any fields on the form have changed

function onSubmit() {
  var field1 = g_form.getControl('caller_id');

  //See if the 'changed' attribute is true
  if(field1.changed) {
    return confirm("The Caller field has changed.\nDo you really want to
save this record?"); } }

Using GlideRecord query from client

There is a lightweight GlideRecord object on the client side that can be used to query the server from within client scripts. The following are some common methods that are available.

• addQuery()
• get()
• hasNext()
• next()
• orderBy()
• query()

For additional information, see GlideRecord() on the ServiceNow Developer Site.

One limitation is that the addOrCondition() method is not available. You can work around this by setting the GlideRecord encodedQuery property with an encoded query string.

Example:

var cat = g_form.getValue('category');
var subcat = g_form.getValue('subcategory');
var lookup = new GlideRecord('u_assignment_lookup');
strQuery = "u_type=Incident^ORu_type=All";
strQuery = strQuery + "^u_category=" + cat;
strQuery = strQuery + "^u_subcategory=" + subcat;
lookup.encodedQuery = strQuery;
lookup.query();

Check if order guide is loaded for the first time

It can be helpful for a catalog client script to know whether the user has just started an order guide or is going back to the guide using the Describe Needs button from the item page.

// onload catalog script
if(g_form.getParameter('cart_edit') != 'guide_serial') {
  // this stuff only runs if the guide is reloaded after initial submit
  // it will not run on initial load of the guide }
Mobile Client GlideForm (g form) scripting

Client scripting for mobile is identical to scripting for the web, with some exceptions. All new scripts must conform to certain guidelines.

For your existing scripts to run correctly on mobile devices, you must modify them to conform to the requirements described in Mobile platform migration impacts on page 3922.

For related information on developing for the mobile interface, see Mobile experience set up for admins on page 1201.

**Mobile platform migration impacts**
The following are affected on the mobile platform: client scripts, UI policies, navigator modules, and UI actions.

### Client scripts

For new or existing scripts to be valid for mobile, they must conform to the following requirements:

- Use the new mobile methods in place of `g_form.getControl()`.
- Do not use deprecated methods.
- Do not reference unsupported browser objects.
- Do not make synchronous JavaScript, GlideAjax, and GlideRecord calls.
- Do not call methods that are not available for mobile.
- Enable scripts to run on the mobile UI.

**Table 1372: Requirements**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the new mobile methods</td>
<td>Several new methods are available for modifying form fields instead of directly manipulating the HTML. These methods replace previous usages of <code>g_form.getControl()</code>, which is deprecated for the mobile platform. In your existing scripts, ensure that the new methods are used in place of methods that are not valid on the mobile platform. For information on these new methods, refer to Mobile GlideForm(). See the ServiceNow Developers site for API information.</td>
</tr>
</tbody>
</table>
| Do not use deprecated methods | The following methods have been deprecated for the mobile platform because direct access to HTML elements is not allowed:

- `g_form.getControl()`
- `g_form.getFormElement()`
- `g_form.getElementById()`

To ensure that existing scripts are compatible, remove all calls to deprecated methods from your code. For new scripts, do not use deprecated methods if you want the script to be valid for mobile.

For `g_form.getControl()`, some of the functionality previously included with this method has been extracted to individual methods. Instead of `g_form.getControl()`, use the new methods described on the developer site.

See the [ServiceNow Developers site](https://developer.servicenow.com) for API information. |
| Do not reference unsupported browser objects | The following browser objects are not supported in mobile scripts:

- `Window`
- `jQuery` or `Prototype` ($, $j, or $$)
- `Document`

Make sure that new scripts do not use these objects, and remove any usage of these objects from your existing scripts. Use `GlideForm` (g_form) instead, which provides methods such as `setLabel()`, `addDecoration()`, and `hasField()` for accomplishing the same tasks. |
| Do not make synchronous JavaScript calls | The mobile platform does not allow synchronous JavaScript calls. The `g_form.getReference()` method must now have the callback parameter defined. For example:

```
g_form.getReference(fieldName, callback)
```

Be sure that all `g_form.getReference()` calls include the callback parameter. For example, the following script does not include a callback and is incompatible with the mobile platform:

```
var userName =
g_form.getReference('assigned_to').user_name,
g_form.setValue('u_assigned_user_name', userName);
```

The following script has been updated to include the callback and is compatible with the mobile platform:

```
g_form.getReference('assigned_to', function(gr) {
    g_form.setValue('u_assigned_user_name', gr.user_name);
});
```

<p>| Do not make synchronous Ajax calls | The mobile platform does not allow synchronous GlideAjax calls. Any use of <code>getXMLWait()</code> in a GlideAjax call will not work on the mobile platform. Be sure that all GlideAjax calls are asynchronous. For more on synchronous versus asynchronous GlideAjax calls and <code>getXMLWait()</code>, see <a href="#">GlideAjax</a>. For information on the available GlideAjax methods, refer to the <a href="#">development portal</a>. |</p>
<table>
<thead>
<tr>
<th>Do not make synchronous GlideRecord calls</th>
<th>The mobile platform does not allow synchronous GlideRecord calls. Make sure that any existing GlideRecord calls include a callback. For example, the following script does not include a callback and is incompatible with the mobile platform:</th>
</tr>
</thead>
</table>
| ```javascript
var gr = new GlideRecord('incident');
gr.addQuery('number',
g_form.getValue('related_incident'));
gr.query();
gr.next();
g_form.setValue('u_related_incident_description',
gr.short_description);
``` | The following script has been updated to include the callback, and is compatible with the mobile platform: |
| ```javascript
var gr = new GlideRecord('incident');
gr.addQuery('number',
g_form.getValue('related_incident'));
gr.query(function(gr) {
  gr.next();
  g_form.setValue('u_related_incident_description',
  gr.short_description);
});
``` | |
<table>
<thead>
<tr>
<th>Do not use methods unavailable on the mobile platform</th>
<th>Due to the limitations and reduced functionality that is imposed by the mobile platform, the following methods are not deprecated but are not available on the mobile platform. If these run on the mobile platform, no action occurs:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• showRelatedList ()</td>
<td>• showRelatedList ()</td>
</tr>
<tr>
<td>• hideRelatedList ()</td>
<td>• showRelatedLists ()</td>
</tr>
<tr>
<td>• hideRelatedLists()</td>
<td>• flash()</td>
</tr>
<tr>
<td>• getSections()</td>
<td>• getParameter()</td>
</tr>
<tr>
<td>• enableAttachments()</td>
<td></td>
</tr>
<tr>
<td>• disableAttachments()</td>
<td></td>
</tr>
<tr>
<td>• setReadOnly()</td>
<td></td>
</tr>
<tr>
<td>(Note that setReadOnly() is available)</td>
<td></td>
</tr>
</tbody>
</table>

| Enable scripts for mobile | Scripts must be enabled for the mobile platform. See Enable client scripts for the smartphone interface on page 1227. |
UI policies

A field is now available on the UI Policy form for scripts to run on a desktop or mobile platform, or both at the same time. Update existing policies so that they apply to either the mobile platform or both. For new scripts, also ensure that the mobile option or both is selected. For more on UI policies for mobile, see Enable UI policies for the smartphone interface on page 1225.

Navigator modules

For existing code, modules must be transferred to either the sys_ui_application or sys_ui_module tables to be available on the mobile platform. When developing new code, be sure that all modules are created in the sys_ui_application or sys_ui_module tables. For more information, see Define a new smartphone module on page 1204.

UI actions

UI actions must be transferred to the sys_ui_ng_action table to appear on the mobile platform. UI action scripts that do not use deprecated methods do not require changes to the script itself. For new UI actions, be sure that they are created in the sys_ui_ng_action table. For more information, see Smartphone interface UI actions on page 1229.

Set client script order

Control the order of execution for your client scripts using the Order field. To avoid having two or more client scripts run concurrently and then conflict, you can add an order for the scripts to run in.

Role required: admin

Adding an order to the client script creates a processing sequence, ordered from lowest to highest number. If two scripts conflict, the client script with the lower number executes first.

1. Navigate to System Definition > Client Script and open an existing client script or click New.
2. Configure the form to include the Order field.
3. Add a number to the order field based on what order you want it to run in relation to other client scripts. Choose a lower number for the script you want to execute first.

AJAX

AJAX (asynchronous JavaScript and XML) is a group of interrelated, client-side development techniques used to create asynchronous Web applications.

AJAX enables web applications to send and retrieve information to and from a server in the background, without impacting the user experience with the displayed web page.

GlideAjax

The GlideAjax class allows the execution of server-side code from the client. GlideAjax calls pass parameters to the script includes, and, using naming conventions, allows the use of these parameters.

Note: This functionality requires a knowledge of JavaScript.

Using GlideAjax:

- Initialize GlideAjax with the name of the script include that you want to use.
- When creating the script include, you must set the name field to be exactly the same as the class name.
- When creating the script include, you must select the **Client callable** check box.
- Specify the parameter `sysparm_name`. GlideAjax uses `sysparm_name` to find which function to use.
- Any extra parameters may be passed in, all of which must begin with `sysparm_`. Avoid using predefined parameter names:
  - `sysparm_name`
  - `sysparm_function`
  - `sysparm_value`
  - `sysparm_type`
- Code is then executed with the `getXML()` or `getXMLWait()` functions

For additional information on GlideAjax, refer to *GlideAjax* in the development portal.

**Examples of asynchronous GlideAjax**

There are two parts to the asynchronous GlideAjax script: client-side and server-side code.

### Hello World: Returning a value from the server

**Client side**

This code runs on the client (the web browser). Create a client script as normal. This sends the parameters to server, which then does the processing. So that the client does not wait for the result, a callback function is used to return the result, passed to the `getXML()` function. (In this case it is called `HelloWorldParse`.)

The `getXMLWait()` function does not need a separate callback function, but this will block the client. If the client-server communication takes a long time (for example on slow networks), the application will seem unresponsive and slow. An example of `getXMLWait()` is in the *Examples of synchronous GlideAjax* on page 3929.

```javascript
var ga = new GlideAjax('HelloWorld');
ga.addParam('sysparm_name', 'helloWorld');
ga.addParam('sysparm_user_name', "Bob");
ga.getXML(HelloWorldParse);

function HelloWorldParse(response) {
    var answer = response.responseXML.documentElement.getAttribute("answer");
    alert(answer); }
```

**Server side**

The server-side code for the above function. Do not create a business rule, but instead navigate to **System Definition > Script Include** and create a new script. Paste in the code below.

---

**Note:** You must set the name of the script include to `HelloWorld`.

- The `sys_script_include` code must extend the `AbstractAjaxProcessor` class and be client-callable.
- Function names starting with "_" are considered private and are not callable from the client.
• Avoid overriding methods of `AbstractAjaxProcessor`, including `initialize`. While it is possible to invoke methods of your superclass object which you have overridden, it is complicated and best avoided altogether.

```javascript
var HelloWorld = Class.create();
HelloWorld.prototype =
  Object.extendsObject(AbstractAjaxProcessor, {
  helloWorld:function() { return "Hello " +
    this.getParameter('sysparm_user_name') + "!"; } ,
  _privateFunction: function() { // this function is not client
callable
}
});
```

This results in an alert box that says 'Hello Bob!' when you visit the form.

**Returning multiple values**

Since the response is an XML document we are not limited to returning a single `answer` value. Here is a more complex example returning multiple XML nodes and attributes.

**AJAX processor script include**

```javascript
/*
* MyFavoritesAjax script include Description - sample AJAX
processor returning multiple value pairs
*/
var MyFavoritesAjax = Class.create();
MyFavoritesAjax.prototype =
  Object.extendsObject(AbstractAjaxProcessor, {

  /*
  * method available to client scripts call using:
  * var gajax = new GlideAjax("MyFavoritesAjax");
  * gajax.addParam("sysparm_name","getFavorites");
  */
  getFavorites: function() { // build new response xml element
    for result
      var result = this.newItem("result");
      result.setAttribute("message","returning all favorites");

    //add some favorite nodes with name and value attributes
    this._addFavorite("color","blue");
    this._addFavorite("beer","lager");
    this._addFavorite("pet","dog");
},

  // all items are returned to the client through the inherited
  methods of AbstractAjaxProcessor
  _addFavorite: function(name, value) {
    var favs = this.newItem("favorite");
    favs.setAttribute("name",name);
    favs.setAttribute("value",value); },

  type:"MyFavoritesAjax"
});
```
Client script

```javascript
// new GlideAjax object referencing name of AJAX script include
var ga = new GlideAjax("MyFavoritesAjax");
// add name parameter to define which function we want to call
// method name in script include will be getFavorites
ga.addParam("sysparm_name","getFavorites");

// submit request to server, call ajaxResponse function with server response
ga.getXML(ajaxResponse);

function ajaxResponse(serverResponse) {
    // get result element and attributes
    var result = serverResponse.responseXML.getElementsByTagName("result");
    var message = result[0].getAttribute("message");

    // check for message attribute and alert user
    if(message) alert(message);

    // build output to display on client for testing
    var output = "";

    // get favorite elements
    var favorites = serverResponse.responseXML.getElementsByTagName("favorite");
    for(var i = 0; i < favorites.length; i++) {
        var name = favorites[i].getAttribute("name");
        var value = favorites[i].getAttribute("value");
        output += name + " = " + value + "\n";
    }
    alert(output);
}
```

XML response

```xml
<xml sysparm_max= "15" sysparm_name="getFavorites"
    sysparm_processor="MyFavoritesAjax">
    <result message = "returning all favorites"></result>
    <favorite name = "color" value = "blue"></favorite>
    <favorite name = "beer" value = "lager"></favorite>
    <favorite name = "pet" value = "dog"></favorite>
</xml>
```

Examples of synchronous GlideAjax

Use synchronous when your script cannot continue without the GlideAjax response. This stops the session until the response is received.

If your use case demands that no further processing can occur until the GlideAjax response has been received, you can use `getXMLWait()`. However, because this will slow down your code and lock the user session until the response is received, it is generally recommended that you use `getXML()` with a callback function.

**Note:** Do not use `AJAXEvaluateSynchronously`.

**Note:** The `getXMLWait()` method is not available in scoped applications.

© 2017 ServiceNow. All rights reserved. 3929
This code results in a client-side alert that displays `The Server Says Hello Bob!`.

The client code.

```javascript
var ga = new GlideAjax('HelloWorld');
ga.addParam('sysparm_name','helloWorld');
ga.addParam('sysparm_user_name','Bob');
ga.getXMLWait();
alert(ga.getAnswer());
```

The server-side script include code.

```javascript
var HelloWorld = Class.create();
HelloWorld.prototype = Object.extendsObject(AbstractAjaxProcessor, {
  helloWorld: function() { return "The Server Says Hello " +
    this.getParameter('sysparm_user_name') + "!"; }
});
```

**AJAXClientHelper**

Provides helper functions for Ajax clients to generate Choice Lists and retrieve displayed values from a choice list.

**Where to Use**

Use this script include wherever you need to generate Choice Lists or retrieve a value from an Ajax client.

**Method Summary**

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getValues()</td>
<td>Gets the values from the Choice List.</td>
</tr>
<tr>
<td>generateChoice()</td>
<td>Generates the choices for a Choice List.</td>
</tr>
<tr>
<td>generateChoiceTable()</td>
<td>Generates the choice table.</td>
</tr>
<tr>
<td>getDisplay()</td>
<td>Gets the display value from the choice list.</td>
</tr>
</tbody>
</table>

**Method Detail**

**getValues()**

Gets the values from the Choice List.

**Input Fields**

Parameters:
- None

**Output Fields**

Returns: The choice list values.

**generateChoice()**

Generates the choices for a Choice List.

**Input Fields**
Parameters:
• None

Output Fields
Returns: The choice list.
generateChoiceTable()
Generates the choice table.

Input Fields
Parameters:
• None

Output Fields
Returns: The choice table.
getDisplay()
Gets the display value from the choice list.

Input Fields
Parameters:
• None

Output Fields
Returns: The display value.

AJAXClientTiming
Saves client timing values for a transaction.

Table 1373: Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>process()</td>
<td>Method called by the Prototype JavaScript Framework during object processing.</td>
</tr>
</tbody>
</table>

Table 1374: Method Detail

<table>
<thead>
<tr>
<th>Method Detail</th>
<th>Description</th>
<th>Input Fields</th>
<th>Output Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>process()</td>
<td>Called by the Prototype JavaScript Framework during object processing. Do not call this method directly.</td>
<td>none</td>
<td>Returns: void.</td>
</tr>
</tbody>
</table>

Jelly tags
This is a list of Jelly tags and Glide tags.
Jelly Tags

if

- **Description:** The *if* tag is just what it looks like, an if tag. This is like an if statement in any programming language, but keep in mind that there is no elseif tag and no else tag. If you want to create that kind of structure, try the choose/when/otherwise syntax.
- **Parameters:** *test* - The condition to evaluate in order to determine if the block will execute.
- **Example:**

  ```
  <g:evaluate var="jvar_gr" object="true">
    var gr = new GlideRecord("incident");
    gr.addQuery("active", true);
    gr.query();
    gr;
  </g:evaluate>
  
  <j:if test="!${jvar_gr.hasNext()}">
    We did not find any active incidents.
  </j:if>
  <j:if test="${jvar_gr.next()}">
    We found ${jvar_gr.getRowCount()} active incidents.
  </j:if>
  ```

while

- **Description:** The *while* tag does a while loop.
- **Parameters:** *test* - The condition to evaluate in order to determine if the statement will loop through. This should be an expression enclosed in ${} or [] that evaluates to true or false.
- **Example:**

  ```
  <g:evaluate var="jvar_gr" object="true">
    var gr = new GlideRecord("incident");
    gr.addQuery("active", true);
    gr.query();
    gr;
  </g:evaluate>
  
  <j:while test="${jvar_gr.next()}">
    <a href="incident.do?sys_id=${jvar_gr.getValue('sys_id')}">${jvar_gr.getValue('number')}</a>
  </j:while>
  ```

set

- **Description:** The *set* tag sets a variable.
- **Parameters:**
- **var** - The variable to set. Often the system prefixes these variables with jvar_ for consistency.
- **value** - The value to set var to. This is often an expression enclosed in ${} or $[].
- **defaultValue** - If the value results to null or empty, this value is put into the var.

**Example:**

```<j:set var="jvar_incident_number" value="${jvar_gr.getValue('number')}"/>```

**set_if**

- **Description**: The set_if tag sets a variable based on a test. This tag is similar to the ternary operator in other programming languages (var = <test> ? <if_true> : <if_false>).
- **Parameters**:
  - **var** - The variable to set. Often the system prefixes these variables with jvar_ for consistency.
  - **test** - The condition to evaluate in order to determine if the statement will evaluate the true value or the false value. This should be an expression enclosed in ${} or $[] that evaluates to true or false.
  - **true** - The value to set the variable to if test evaluates to true. This parameter is optional, so if the field is blank, and if test evaluates to true, the variable will be left blank.
  - **false** - The value to set the variable to if test evaluates to false. This parameter is optional, so if the field is blank, and if test evaluates to false, the variable will be left blank.

**choose**

- **Description**: The choose tag starts a choose block of code. This is similar to the if-elseif-else kind of syntax in most programming languages. With a choose tag, you can use when and otherwise tags to specify other blocks of code.
- **Parameters**: None.
- **Example**:

```<j:choose>
  <j:when test="${jvar_gr.getRowCount() lt 1}">We found multiple records!</j:when>
  <j:when test="${jvar_gr.next()}">We```
When

- **Description**: The `when` tag is used within a choose block to indicate a condition. This tag is similar to an if or an elseif in that it specifies a condition, executes the inner content, and then implies a break at the end to leave the if-elseif construct.

- **Parameters**: `test` - The condition to evaluate in order to determine if the statement will loop through. This should be an expression enclosed in `{}` or `[]` that evaluates to true or false.

- **Example**:

```xml
<j:choose>
  <j:when
    test="${jvar_gr.getRowCount()} ${AMP}lt; 1">We found multiple records!</j:when>
  <j:when
    test="${jvar_gr.next()}">We found record
    ${jvar_gr.getValue('number')}\</j:when>
  <j:otherwise>Sorry, we could not find the record you specified.</j:otherwise>
</j:choose>
```

Otherwise

- **Description**: The `otherwise` tag is used within a choose/when/otherwise block, and is like the "else" or "default" case.

- **Parameters**: None.

- **Example**:

```xml
<j:choose>
  <j:when
    test="${jvar_gr.getRowCount()} ${AMP}lt; 1">We found multiple records!</j:when>
  <j:when
    test="${jvar_gr.next()}">We found record
    ${jvar_gr.getValue('number')}\</j:when>
  <j:otherwise>Sorry, we could not find the record you specified.</j:otherwise>
</j:choose>
```
Glide Tags

**evaluate**

- **Description:** The `evaluate` tag evaluates JavaScript code (server side), and makes variables visible to future code. Unlike other tags, the evaluate tag evaluates the content that is inside the tag as server side JavaScript.

  The context is the same as that of script includes in the system. Other script includes, global business rules, GlideRecord, GlideSystem, and Jelly variables (prefixed with jelly, if the parameter jelly="true" is set) are available.

- **Parameters:**
  - `var` - The name of the variable that will be set to the result of the script.
  - `object` - If set to true, the result of the expression will be treated as an object instead of a primitive variable (string or integer variable values).
  - `jelly` - If set to true, allows Jelly context variables to be referenced in the script.
  - `expression` - This is an expression to be executed for the value to put in `var`. The expression can be either of two places. First, it can be an attribute on the evaluate tag itself. Otherwise, the content between the beginning tag and ending tag is the expression. The last line of the expression is the actual value passed into `var`.

- **Example:**

  ```
  <g:evaluate var="jvar_gr" object="true">
    var gr = new GlideRecord("incident");
    gr.addQuery("active", "true");
    gr.query();
    gr; // this is the variable put into the variable jvar_gr
  </g:evaluate>
  ```

**messages**

- **Description:** The `messages` tag helps with translation. When `gs.getMessage()` is called anywhere on a page, there are two possible
places where the translation is found. First, the page checks a local cache of translations. Second, the page makes an AJAX call to the server to find the translation. What g:messages does is allow pages to cache certain messages.

- **Parameters**: None.
- **Example**:

```html
<g:messages>
  Yes
  No
  Cancel
</g:messages>
```

**breakpoint**

- **Description**: When the `breakpoint` tag is called, it prints a list of all the variables in Jelly at the current moment, with their respective values. If a variable is specified, it prints the requested variable and its value. The output is placed in the System Log.
- **Parameters**: `var` - (Optional) The variable to log the value for. If `var` is not specified, then all variables will be dumped into the log.
- **Example**:

```html
<g:breakpoint />
```

```html
<g:breakpoint var="sysparm_view"/>
```

**no_escape**

- **Description**: The system, by default, uses escaped output as a security measure. Output placed inside of `no_escape` tags is not escaped before output. Be careful when using these tags, because if user input is displayed here it can open a security vulnerability on the page.
- **Parameters**: None.
- **Example**:

```html
<g:no_escape>
${jvar_raw_html_data}
</g:no_escape>
```

**macro_invoke**

- **Description**: The `macro_invoke` tag calls a UI macro that you have specified in the database. You may also call a UI macro by specifying it in the tag name. For example, if you had a UI macro named `my_macro`, you could call that macro with the tag `<g:my_macro/>`.
- **Parameters**:
  - `macro` - The name of the UI macro to execute. If your tag name is `g:macro_invoke`, then the macro attribute specifies the name
of the macro. If the tag name includes the name of the macro, then there is no need to include a macro attribute.

- Other attributes - For each attribute you specify, a variable with that name will be available in the context of the UI macro, prefixed with "jvar_".

- Example:

```xml
<!-- Will invoke the contents of the UI macro named "sample_macro", which will have the variable jvar_message available within it-->
<g:macro_invoke
macro="sample_macro"
message="This is a sample macro variable." />

<!-- Will invoke the contents of the UI macro named "sample_macro", which will have the variable jvar_message available within it-->
<g:sample_macro message="This is a sample macro variable." />
```

Extensions to Jelly syntax

Apache's Jelly syntax is used to render forms, lists, UI pages, and many other things rendered in ServiceNow.

With Jelly, logic can be embedded within static content and computed values may be inserted into the static content.

**Attention:** This functionality requires a knowledge of Apache Jelly (a Java and XML based scripting and processing engine for turning XML into executable code).

This page from Apache has a summary of the standard Jelly tags: [http://commons.apache.org/jelly/tags.html](http://commons.apache.org/jelly/tags.html)


**Namespaces**

Jelly often includes multiple namespaces when invoking tags.

The "j" namespaces are standard Jelly whereas the "g" namespaces are unique to ServiceNow scripts. For example, the `<g:evaluate>` tag is supplied by ServiceNow to allow you to compute a value using JavaScript. The standard Jelly tag `<j:test>` is used to evaluate a condition.

**Phases**

Usually, there are two phases indicated by namespaces `<j>` versus `<j2>` and `<g>` versus `<g2>`.

The namespaces without the "2" happen in the first phase of processing and these are cached except when used in a UI page. Those with the "2" are never cached. Care must be taken when selecting whether to use phase 1 or phase 2 for efficiency and correct results.
In addition to the namespaces, the syntax used to insert values into static content differs depending on which phase is to supply the value. A dollar with braces surrounding a value inserts the value in phase 1. For example, ${jvar_ref} inserts the value jvar_ref during phase 1 of the jelly process. A dollar with brackets surrounding a value inserts the value in phase 2. For example, ${jvar_ref} inserts the value jvar_ref during phase 2. A value surrounded by quotes is treated as a string. For example, '\${jvar_ref}' inserts the value jvar_ref as a string during phase 2.

```javascript
if (confirm("$[gs.getMessage('home.delete.confirm')]"))
...
</script>
```

```html
<input type="hidden" id="${jvar_name}" name="${jvar_name}" value="${jvar_value}" class="${jvar_class}" />
```

**If tests**

Testing whether something is true or not can be done as follows:

```html
<j:if test="${jvar_something}">...do something...</j:if>
<j:if test="${!jvar_something}">...do something...</j:if>
```

The reason this works, is that, in Jelly, a term like jvar_something is "truthful" in an if tag if:
1. it is Boolean and true
2. it is a String and = "true", "yes", "on", or "1"

Testing whether something exists can be done as follows:

```html
<j:if test="${empty(jvar_something)}">...do something...</j:if>
```

The reason this works is that the jexl empty function returns true if its argument is:
1. null
2. an empty string
3. a zero length collection
4. a map with no keys
5. an empty array

**Set_IF**

Sets a variable to one of two different values depending on whether a test is true or false.

```html
<g2:set_if var="jvar_style" test="$[gs.getPreference('table.compact') != 'false']" 
  true="margin-top:0px; margin-bottom:0px;" 
  false="margin-top:2px; margin-bottom:2px;" />
```

**<g:insert> versus <g:inline> versus <g:call>**

This page provides a comparative explanation of three tags: <g:insert>, <g:inline>, and <g:call>. 

© 2017 ServiceNow. All rights reserved. 3938
<g:insert>
The <g:insert> tag inserts a Jelly file into your Jelly in a new context. This means you cannot access the variables previously established in your Jelly.

<g:insert template="get_target_form_function.xml" />

<g:inline>
The <g:inline> tag inserts a Jelly file into your Jelly in the same context. This means that the inserted Jelly can access the variables you previously established and it can change the values of those variables.

<g:inline template="element_default.xml" />

<g:call>
For better encapsulation, the <g:call> tag may be used. Your function will only have access to the values passed to it. The Jelly context will look the same after a call as before the call. This means you cannot set a global variable here and read it later. This also means you can't mistakenly set a global variable called "jvar_temp" and overwrite a variable that somebody else was relying on.

Passing values, if needed, is done explicitly by including the name of the parameter on the <g:call> line followed by the equal sign followed by the value in quotes:

<g:call function="collapsing_image.xml" id="${jvar_section_id}" image="${jvar_cimg}" first_section_id="${jvar_first_section_id}" image_alt="${jvar_cimg_alt}" />

If values are passed, and you want to have defaults or required parameters, your Jelly referenced in the function must then include a line to declare whether the parameters are required or have a default value:

<g:function id="REQUIRED" image="REQUIRED" image_prefix="" image_alt="REQUIRED"/>

The example above indicates that 3 of the parameter are required and one parameter is option with a blank default value. Note that if you are not passing values or if you do want to have default or required values, you do not need to include the <g:function> line at all. In general, however, you will want to include a <g:function> line.

The value can then be referenced in your template by prepending the "jvar_" prefix to the parameter's name:

<img id="img.${jvar_id}" src="images/${jvar_image}" alt="${jvar_image_alt}" onclick="toggleSectionDisplay('${jvar_id}','${jvar_image_prefix}','${jvar_first_section_id}');"/>

For <g:call>, parameters may also be pass implicitly as a list of named variables in an "arguments" parameter:

<g:call function="item_link_default.xml" arguments="sysparm_view,ref_parent,jvar_target_text"/>

© 2017 ServiceNow. All rights reserved.  3939
As an alternative to passing variables into the function via separate tag arguments, it is possible to pass a list of variables in a single 'arguments' argument. All variables identified by name (comma separated) in the argument parameter are re-introduced within the function under the exact same name (e.g. inside the function template, we'd have variables sysparm_view, ref_parent, and jvar_target_text available to us).

The function template may return a value to the calling template using the return attribute. Within the function the jvar_answer variable sets the return value.

```xml
<g:call function="item_body_cell_calc_style.xml" arguments="jvar_type" return="jvar_style"/>
```

The <g:evaluate> tag is used to evaluate an expression written in Rhino JavaScript and sometimes to set a variable to the value of the expression.

The last statement in the expression is the value the variable will contain.

```xml
<g2:evaluate var="jvar_page" jelly="true">
  var page = "";
  var pageTitle = "";
  var pageGR = new GlideRecord("cmn_schedule_page");
  pageGR.addQuery("type", jelly.jvar_type);
  pageGR.query();
  if (pageGR.next()) {
    page = pageGR.getValue("sys_id");
    pageTitle = pageGR.getDisplayValue();
  }
  page;
</g2:evaluate>
```

If you would like to have the evaluate return an object (for example an array), use the argument object="true".

```xml
<g2:evaluate object="true" var="jvar_items" expression="SncRelationships.getCMDBViews()" />
```

If you would like to access Jelly variables inside an evaluate, include jelly="true" in the evaluate and add "jelly." before the Jelly variable's name. For example, to access the GlideJellyContext:

```xml
<g2:evaluate var="jvar_row_no" jelly="true">
  var gf = jelly.context.getGlideForm();
  var row = gf.getRowNumber();
  row;
</g2:evaluate>
```

Another example of accessing a jvar using the jelly="true" parameter. The value of jvar_h was set previously and we can access it inside the evaluate:

```xml
[$[NLBR:jvar_h.getHTMLValue('newvalue')]
<g2:evaluate var="jvar_fix_escaping" jelly="true">
```

© 2017 ServiceNow. All rights reserved.  3940
var auditValue = jelly.jvar_h.getHTMLValue('newvalue');
gs.log("************ " + auditValue);
</g2:evaluate>

copyToPhase2="true"

If you have a need to take the results of an evaluation that occurs in phase 1 and propagate it to phase 2, use copyToPhase2="true". There is some protection for escaping in this use. For example:

```
<g:evaluate var="jvar_has_special_inc" copyToPhase2="true">
    var specialInc = gs.tableExists("special_incident");
    specialInc;
</g:evaluate>
[jvar_has_special_inc]
```

If you do not need to evaluate something, you can do this more directly. Beware of escaping issues here (double quotes in jvar_rows would cause a problem in the example):

```
<j2:set var="jvar_rows" value="${jvar_rows}"/>
```

This tag can be used to display the current Jelly variables and their values in the log.

Be sure to remove this tag before going to production.

```
<g:include_script/>
```

In pre-Fuji releases, this tag can be used to include UI scripts for client side access to reusable javascript.

For example, the jQuery library is included as a UI script. To view this library, navigate to System UI > UI Scripts > jquery.min. To include this library in a UI page or macro, add `<g:include_script src="jquery.min.jsdbx"/>` to the jelly script. This example uses the name of the UI script with the extension .jsdbx appended.

Starting with the Fuji release, you cannot use this include or you might break the existing jQuery code on the instance.

```
<g:ui_form/>
```

This tag defines a form on the UI page.

For example, if your form contained the application_sys_id field:

```
<g:ui_form>
    <p>Click OK to run the processing script.</p>
    <g:dialog_buttons_ok_cancel ok="return true" />
    <input type="hidden" name="application_sys_id" value="499836460a0a0b1700003e7ad950b5da"/>
</g:ui_form>
```

The g:ui_form may benefit greatly from a processing script.

```
<g:ui_input_field />
```

This tag adds a reference to a UI macro that creates an input field on a page that allows users to input information. The ui_input_field passes a label, name, value, and size into the UI macro.

Here is an example from a UI page:

```
<g:ui_input_field label="sys_id" name="sysid" value="9d385017c611228701d22104cc95c371" size="50"/>
```
This tag puts a user-editable check mark on a page. The name and value are passed into the UI macro.

Here is an example from a table on a UI page:

```
<table>
  <tr>
    <td nowrap="true">
      <label>Time Card Active:</label>
    </td>
    <td>
      <g:ui_checkbox name="timecard_active" value="${sysparm_timecard_active}"/>
    </td>
  </tr>
</table>
```

This tag puts buttons on the UI page that run a specified processing script if the tag returns true.

If your UI page contains a form (uses the `<g:form>` tag), you can submit the form and have the Processing Script run. The Processing Script can naturally access fields on the form. For example, if your form contained the `application_sys_id` field:

```
<g:ui_form>
  <p>Click OK to run the processing script.</p>
  <g:dialog_buttons_ok_cancel ok="return true" />
  <input type="hidden" name="application_sys_id" value="499836460a0a0b1700003e7ad950b5da"/>
</g:ui_form>
```

This tag adds a reference to a page that can be referenced by a Processing Script.

The following example creates a reference defined by name, id, and table parameters in the tag:

```
<g:ui_reference name="QUERY:active=true^roles=itil" id="assigned_to" table="sys_user" />
```

Then in the Processing Script, reference the name field like this:

```
newTask.assigned_to = request.getParameter("QUERY:active=true^roles=itil");
```

You can specify a reference qualifier, so that the "name" attribute can be unique. The following example creates a reference defined by name, id, and table parameters in the tag. Note: the "columns" attribute only applies to the auto-completer.

```
<g:ui_reference name="parent_id" id="parent_id" table="pm_project"
  query="active=true" completer="AJAXTableCompleter" columns="project_manager;short_description"/>
```

**Ampersand**

Ampersands in Jelly can cause you grief because Jelly is XML.

Use `${AMP}` to insert an ampersand in Jelly. If you are writing JavaScript that appears in the HTML part of say a UI page or UI macro that is actually going to run on the browser you are better off putting this code in the "client script" field and that way you can avoid escaping issues. However, if you really must put it in the "HTML" field, you will need to do something like this:

```
ta = ta[1].split('${AMP}');
```
And
Use ${AMP} to insert a JavaScript and in Jelly.

For example:

```javascript
if (d ${AND} e)
    var color = d.value;
```

Alternately, in a Jelly test you would use &amp;amp. For example:

```xml
<j:if test="${jvar_form_name == 'sys_form_template' && !RP.isDialog()}">

Less than
Similar to ampersands, less than ("<") signs can also cause problems due to Jelly being XML. This can be resolved by reversing your test such that it is not necessary or by using ${AMP}lt; in place of the less than sign.

```xml
<g2:evaluate var="jvar_text">
    var days = "";
    var selectedDays = '${${ref}}';
    for (var i = 1; i <= 7; i++) {
        if (selectedDays.indexOf(i.toString()) >= 0) {
            days += gs.getMessage("dow" + i);
            days += " ";
        }
    }
    days;
</g2:evaluate>
```

Many times you can avoid the "less than" operator altogether by just using "not equals" which doesn't have escaping issues. For example:

```javascript
for (var i=0; i != ta.length; i++) {
}
```

Whitespace
Normally, white space is removed by Jelly. To keep it, you must specify that it not be trimmed.

For example, the following keeps the space after the colon.

```xml
<j2:whitespace trim="false">${gs.getMessage('Did you mean')}: ${SP}</j2:whitespace>
```

Spaces
To encode a non-breaking space (&nbsp;), you can use ${SP}.

For example:

```html
<span id="gsft_domain" style="display: inline">
    ${gs.getMessage('Domain')} ${SP}
    <span id="domainDD" class="drop_down_element" style="text-decoration: none; color: white">
        ${gs.getMessage('Loading...')}
    </span>
</span>
```

Tracing Jelly
ServiceNow has a feature that allows the evaluation of Jelly to be traced.
The trace is sent to the log. This should only be turned on during debugging as this produces a lot of logging. To turn on the trace, set the property glide.ui.template.trace to true. For example, the following script can be executed to do this:

```java
GlideProperties.set ( 'glide.ui.template.trace', true );
```

If you want to see your log entries on your web browser at the bottom of each page, navigate to **System Diagnostics > Debug Log**.

**Jelly escaping types**

**Note:** This functionality requires a knowledge of JavaScript, HTML, and Apache Jelly (a Java and XML based scripting and processing engine for turning XML into executable code).

There are two different types of escaping that is required when generating output from Jelly:

- JavaScript
- HTML

The escaping for each of these consists of:

<table>
<thead>
<tr>
<th>Type</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript</td>
<td>' (single quote)</td>
<td>&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot; (double quote)</td>
<td>&quot;</td>
</tr>
<tr>
<td></td>
<td>CR (carriage return)</td>
<td>(blank)</td>
</tr>
<tr>
<td></td>
<td>NL (newline)</td>
<td>\n (' followed by 'n')</td>
</tr>
<tr>
<td>HTML</td>
<td>&amp; (ampersand)</td>
<td>&amp;</td>
</tr>
<tr>
<td></td>
<td>&lt; (less than)</td>
<td>&lt;</td>
</tr>
<tr>
<td></td>
<td>&gt; (greater than)</td>
<td>&gt;</td>
</tr>
</tbody>
</table>

You can also escape HTML using the `getHTMLValue()` function which will enforce all line breaks and escape the characters mentioned above. It can be used as follows:

```java
${test.getHTMLValue()}
```

Add escaping to a Jelly replacement

You can handle character escaping in Jelly files. XML escaping behavior can be modified only by users with the security admin role.

**Note:** This functionality requires a knowledge of JavaScript, HTML, and Apache Jelly (a Java and XML based scripting and processing engine for turning XML into executable code).

Add a prefix to the `_${expression}_` or `_${expression}_` indicating the escaping to be performed.

```java
${JS:expression}
${HTML:expression}
```
The prefix tells the system to take the result of the expression and escape it before outputting. The escaping may be combined by specifying a comma-separated list of prefixes:

$$\{JS,HTML:expression\}$$

Client scripts good practices

Follow these practices to help ensure that client scripts work effectively and efficiently.

A client script is a piece of JavaScript code that runs on the client, rather than the server. With the exception of onCellEdit client scripts, client scripts apply to forms only.

Well-designed client scripts can reduce the amount of time it takes to complete a form. However, improperly implemented client scripts can significantly slow down form load times.

Restrict list editing

If you create UI policies or client scripts for fields on a form, you must use another method to ensure that data in those fields is similarly controlled in a list.

With the exception of onCellEdit client scripts, UI policies and client scripts apply to forms only. Use the following methods to restrict list editing when using client scripts:

- Disable list editing for the table.
- Create appropriate business rules or access controls for list editing.
- Create data policies.
- Create a separate onCellEdit client script.

Minimize server lookups

Use client data as much as possible to eliminate the need for time-consuming server lookups.

Client scripting uses either data available on the client or data retrieved from the server. The top ways to get information from the server are g_scratchpad and asynchronous GlideAjax lookup.

The primary difference between these methods is that g_scratchpad is sent once when a form is loaded (information is pushed from the server to the client), whereas GlideAjax is dynamically triggered when the client requests information from the server.

**Note:** GlideRecord and g_form.getReference() are also available for retrieving server information. However, these methods are no longer recommended due to their performance impact. Both methods retrieve all fields in the requested GlideRecord when most cases only require one field.

Example - retrieve server data using g_scratchpad

The g_scratchpad object passes information from the server to the client, such as when the client requires information not available on the form.

For example, if you have a client script that needs to access the field u_retrieve, and the field is not on the form, the data is not available to the client script. A typical solution to this situation is to place the field on the form and then always hide it with a client script or UI policy. While this solution may be faster to configure, it is slower to execute.

If you know what information the client needs from the server before the form is loaded, a display business rule can create g_scratchpad properties to hold this information. The g_scratchpad is sent to the client when the form is requested, making it available to all client-side scripting methods. This is a very efficient means of sending information from the server to the client. However, you can only load data this way when the form is loaded. The business rule cannot be triggered dynamically. In those cases, use an asynchronous GlideAjax call.
For example, assume you open an incident and need to pass this information to the client:

- The value of the system property css.base.color
- Whether or not the current record has attachments
- The name of the caller's manager

A display business rule sends this information to the client using the following script:

```javascript
// Get the value of the system property css.base.color
g_scratchpad.css = gs.getProperty('css.base.color');

// Check if the current record has attachments
if (g_scratchpad.hasAttachments)
  // do something interesting here
else
  alert('You need to attach a form signed by ' +
        g_scratchpad.managerName);
```

Example - retrieve server data using asynchronous GlideAjax

Asynchronous GlideAjax allows you to dynamically request information from the server.

This script compares the support group of the CI and the assignment group of the incident by name:

```javascript
// Check if the assignment group's name matches the support group
function onChange(control, oldValue, newValue, isLoading) {
  if (isLoading)
    return;

  var ga = new GlideAjax('ciCheck');

  ga.addParam('sysparm_name', 'getCiSupportGroup');
  ga.addParam('sysparm_ci', g_form.getValue('cmdb_ci'));
  ga.addParam('sysparm_ag', g_form.getValue('assignment_group'));

  ga.getXML(doAlert); // Always try to use asynchronous (getXML) calls rather than synchronous (getXMLWait)
}

// Callback function to process the response returned from the server
function doAlert(response) {
  var answer = response.responseXML.documentElement.getAttribute("answer");

  alert(answer);
}
```

This script relies on the accompanying script include:

```javascript
var ciCheck = Class.create();
```
Use the setValue() displayValue parameter for reference fields

When using setValue() on a reference field, include the displayValue parameter to avoid additional server calls.

When using setValue() on a reference field, be sure to include the reference field display value as the 3rd parameter. If you set the value without the displayValue, the instance does a synchronous call to retrieve the display value for the record you specified. This extra round trip to the server can impact performance.

This example demonstrates the incorrect way to call setValue:

```javascript
var id = '5137153cc611227c000bbd1bd8cd2005';
g_form.setValue('assigned_to', id); // Client needs to go back to the server to fetch the name that goes with this ID
```

Instead, include the display value as an optional parameter in setValue():

```javascript
var id = '5137153cc611227c000bbd1bd8cd2005';
var name = 'Fred Luddy';
g_form.setValue('assigned_to', id, name); // No server call required
```
Use UI policy instead of a client script

When possible, consider using a UI policy instead of a client script.

UI policies provide these benefits over client scripts:

- UI policies have an **Order** field to allow full control over the order in which client-side operations take place.
- UI policies do not require scripting to make a field mandatory, read-only, or visible.

**Note:** UI policies apply after client scripts.

Validate input using client scripts

An excellent use for client scripts is validating input from the user.

This validation improves the user experience because the user finds out if there are data issues before submitting the information.

An example of validation is to verify that the **Impact** field value is valid with the **Priority** field value. In this example, **Low** impact is not allowed with **High** priority.

```javascript
if (g_form.getValue('impact') == '3' &&
g_form.getValue('priority') == '1')
    g_form.showFieldMsg('impact', gs.getMessage('Low impact now allowed with High priority'), 'error');
```

Run only necessary scripts

To avoid running time-consuming scripts unnecessarily, make sure that client scripts perform only necessary tasks.

The following examples demonstrate improvements to the initial code sample. Each example demonstrates a particular enhancement to the script to improve performance and avoid unnecessary calls.

Remember that client scripts have no **Condition** field. This means that onData() and onChange() scripts run in their entirety every time the appropriate form is loaded. This example is an inefficient onChange() client script set to run when the **Configuration item** field changes.

```javascript
//Set Assignment Group to CI's support group if assignment group is empty
function onChange(control, oldValue, newValue, isLoading) {

    var ciSupportGroup =
        g_form.getReference('cmdb_ci').support_group;

    if (ciSupportGroup != '' &&
        g_form.getValue('assignment_group') != '')
        g_form.setValue('assignment_group',
            ciRec.support_group.sys_id);
}
```
This example improves upon the first by replacing the getReference() or GlideRecord lookup with an asynchronous GlideAjax call.

```
//Set Assignment Group to support group if assignment group is empty
function onChange(control, oldValue, newValue, isLoading) {
    var ga = new GlideAjax('ciCheck');
    ga.addParam('sysparm_name', 'getSupportGroup');
    ga.addParam('sysparm_ci', g_form.getValue('cmdb_ci'));
    ga.getXML(setAssignmentGroup);
}
function setAssignmentGroup(response) {
    var answer = response.responseXML.documentElement.getAttribute("answer");
    g_form.setValue('assignment_group', answer);
}
```

The isLoading flag is the simplest way to prevent unnecessary code from taking up browser time in onChange scripts. The isLoading flag should be used at the beginning of any script that is not required to run when the form is loading. There is no need to run this script on a form load because the logic would have already run when the field was last changed. Adding the isLoading check to the script prevents it from doing a cmdb_ci lookup on every form load.

The isTemplate flag indicates that a template is loading.

```
//Set Assignment Group to CI's support group if assignment group is empty
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
    if (isLoading)
        return;
    var ga = new GlideAjax('ciCheck');
    ga.addParam('sysparm_name', 'getSupportGroup');
    ga.addParam('sysparm_ci', g_form.getValue('cmdb_ci'));
    ga.getXML(setAssignmentGroup);
}
function setAssignmentGroup(response) {
    var answer = response.responseXML.documentElement.getAttribute("answer");
    g_form.setValue('assignment_group', answer);
}
```

If the onChange script should run during loading, use the following convention:

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
```

if (isLoading) {}; // run during loading
   // rest of script here
}

The newValue check tells this script to continue only if there is a valid value in the relevant field. This prevents the script from running when the field value is removed or blanked out. This also ensures that there will always be a valid value available when the rest of the script runs.

//Set Assignment Group to CI's support group if assignment group is empty
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
    if (isLoading)
        return;

    if (newValue) {
        var ga = new GlideAjax('ciCheck');

        ga.addParam('sysparm_name', 'getSupportGroup');
        ga.addParam('sysparm_ci', g_form.getValue('cmdb_ci'));
        ga.getXML(setAssignmentGroup);
    }
}

function setAssignmentGroup(response) {
    var answer = response.responseXML.documentElement.getAttribute("answer");

    g_form.setValue('assignment_group', answer);
}

To have the script react to a value that changes after the form loads, use the newValue != oldValue check.

//Set Assignment Group to CI's support group if assignment group is empty
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
    if (isLoading)
        return;

    if (newValue) {
        if (newValue != oldValue) {
            var ga = new GlideAjax('ciCheck');

            ga.addParam('sysparm_name', 'getSupportGroup');
            ga.addParam('sysparm_ci', g_form.getValue('cmdb_ci'));
            ga.getXML(setAssignmentGroup);
        }
    }
}
In this example, the GlideAjax call is buried one level deeper by rearranging the script to check as many things available to the client as possible before running the server calls. The script checks the assignment before executing the GlideAjax call. This prevents the server lookup when the assignment_group field is already set.

```javascript
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
    if (isLoading)
        return;

    if (newValue) {
        if (newValue != oldValue) {
            if (g_form.getValue('assignment_group') == '') {
                var ga = new GlideAjax('ciCheck');
                ga.addParam('sysparm_name', 'getSupportGroup');
                ga.addParam('sysparm_ci', g_form.getValue('cmdb_ci'));
                g_form.getValue('cmdb_ci'));
                ga.getXML(setAssignmentGroup);
            }
        }
    }
}

function setAssignmentGroup(response) {
    var answer = response.responseXML.documentElement.getAttribute("answer");
    g_form.setValue('assignment_group', answer);
}
```

Server API reference

Server-side Glide APIs enable you to create scripts to run on the server. Script includes provide common functionality that you can include in your server-side scripts. This page provides links to reference information for the server-side APIs and script includes.
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMDBUtil</td>
<td>Provides utility methods for creating and managing table relationships in the configuration management database (CMDB) and managing CMDB baselines.</td>
</tr>
<tr>
<td>ExpenseLine</td>
<td>Provides methods for generating expense line (fm_expense_line) records from your own server-side scripts for Cost Management.</td>
</tr>
<tr>
<td>GlideAggregate</td>
<td>Extends GlideRecord to allow database aggregation queries, such COUNT, SUM, MIN, MAX, and AVG, for creating customized reports or calculations in calculated fields.</td>
</tr>
<tr>
<td>GlideDateTime</td>
<td>Performs date-time operations, such as date-time calculations, formatting a date-time, or converting between date-time formats.</td>
</tr>
<tr>
<td>GlideElement</td>
<td>Provides methods to operate on the fields of the current GlideRecord.</td>
</tr>
<tr>
<td>GlideRecord</td>
<td>Provides methods for performing database operations. GlideRecord is a special Java class that can be used in JavaScript as if it were a native JavaScript class, instead of writing SQL queries.</td>
</tr>
<tr>
<td>GlideRecordSecure</td>
<td>Provides GlideRecord methods that enforce access control lists (ACLs).</td>
</tr>
<tr>
<td>GlideSystem</td>
<td>Provides methods to obtain information about the system.</td>
</tr>
<tr>
<td>GlideTimeline</td>
<td>Provides methods for configuring and displaying a Glide Windowing Toolkit Timeline when customizing timeline schedule pages.</td>
</tr>
<tr>
<td>Service Catalog</td>
<td>Provides methods for accessing service catalog actions.</td>
</tr>
</tbody>
</table>
Table 1376: Script Includes reference

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Collision</td>
<td>Provides helper functions useful for scripting Change Collision on the server-side or when using AJAX calls on the client.</td>
</tr>
<tr>
<td>Discovery</td>
<td>Provides functions to accomplish common Discovery tasks.</td>
</tr>
<tr>
<td>JavaScript Tools</td>
<td>Provides general JavaScript utility functions that are useful for tasks such as logging, validating scripts, and exception wrapping.</td>
</tr>
<tr>
<td>MID Server</td>
<td>Provides functions to accomplish common MID Server tasks.</td>
</tr>
<tr>
<td>Utility</td>
<td>Provides general utility functions that are useful when working with arrays, datetimes, and tables.</td>
</tr>
<tr>
<td>Workflow</td>
<td>Provides functions that are useful when working with workflows.</td>
</tr>
</tbody>
</table>

Table 1377: Notify API reference

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify API</td>
<td>Provides functionality for sending SMS messages and setting up conference calls.</td>
</tr>
</tbody>
</table>

Glide class overview

The ServiceNow Glide classes expose JavaScript APIs that enable you to conveniently work with tables using scripts.

Using the Glide APIs, you can perform database operations without writing SQL queries, display UI pages, and define UI actions. The following tables provide brief descriptions of the Glide classes and links to detailed information.

Table 1378: Server-side Glide classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideRecord</td>
<td>Use this class for database operations instead of writing SQL queries. GlideRecord is a special Java class that can be used in JavaScript exactly as if it were a native JavaScript class. A GlideRecord is an object that contains records from a single table. See GlideRecord.</td>
</tr>
<tr>
<td>GlideElement</td>
<td>Use this class to operate on the fields of the current GlideRecord. See GlideElement.</td>
</tr>
<tr>
<td>GlideSystem</td>
<td>Use this class to get information about the system. See GlideSystem.</td>
</tr>
</tbody>
</table>
### Table 1379: Client-side Glide Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlideAggregate</td>
<td>Use this class to perform database aggregation queries, such as COUNT, SUM,</td>
</tr>
<tr>
<td></td>
<td>MIN, MAX, and AVG, for creating customized reports or calculations in</td>
</tr>
<tr>
<td></td>
<td>calculated fields. See <a href="#">GlideAggregate</a>.</td>
</tr>
<tr>
<td>GlideDateTime</td>
<td>Use this class to perform date-time operations, such as date-time</td>
</tr>
<tr>
<td></td>
<td>calculations, formatting a date-time, or converting between date-time</td>
</tr>
<tr>
<td></td>
<td>formats. See <a href="#">GlideDateTime</a>.</td>
</tr>
</tbody>
</table>

### Glide stack

Glide is an extensible Web 2.0 development platform written in Java that facilitates rapid development of forms-based workflow applications (work orders, trouble ticketing, and project management, for example).

### Table 1380: User interface stack technology map

<table>
<thead>
<tr>
<th>Java packages</th>
<th>Technologies used</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface (Browser)</td>
<td><a href="#">AngularJS</a>, <a href="#">HTML</a>, <a href="#">CSS</a>, <a href="#">JavaScript</a></td>
</tr>
<tr>
<td>com.glide.ui</td>
<td>GlideServlet</td>
</tr>
<tr>
<td>com.glide.jelly</td>
<td>Apache Jelly</td>
</tr>
<tr>
<td>com.glide.script</td>
<td>Business Rules</td>
</tr>
<tr>
<td></td>
<td>Mozilla Rhino</td>
</tr>
</tbody>
</table>
### Table 1381: User interface stack technology map descriptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| GlideServlet    | The primary driver of Glide, and the only servlet in the system, is found in GlideServlet.java. The GlideServlet: | • Handles inbound action requests  
• Renders pages  
• Merges data with forms  
• Presents to user  
• Interfaces with script layer |
| Business Rules  |                                                                             | • ECMA / JavaScript implementation based on Mozilla Rhino  
• Interfaces with persistence layer using JDBC recordset interface  
• Merges persistence layer meta-data with data for easy correlation |
| Persistence     |                                                                             | • Persistence means any store  
• RDBMS  
• LDAP  
• File system  
• Uniform access regardless of store type  
• Provides QUID and meta-data capabilities  
• Interfaces presented to callers  
• RecordSet  
• TableDescriptor  
• ElementDescriptor |
Scripting alert, info, and error messages

You can send messages to customers as alerts, informative messages, or error messages.

Table 1382: Business rule and other general use scripts

<table>
<thead>
<tr>
<th>Script</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>current.field_name.setError(&quot;Hello World&quot;);</code></td>
<td>Will put &quot;Hello World&quot; below the specified field.</td>
</tr>
<tr>
<td><code>gs.addInfoMessage(&quot;Hello World&quot;);</code></td>
<td>Will put &quot;Hello World&quot; on the top of the screen.</td>
</tr>
<tr>
<td><code>gs.print(&quot;Hello World&quot;);</code></td>
<td>Will write to the text log on the file system but not to the sys_log table in the database.</td>
</tr>
<tr>
<td>Script</td>
<td>Result</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><code>gs.log(&quot;Hello World&quot;);</code></td>
<td>Will write to the database and the log file.</td>
</tr>
</tbody>
</table>

**Note:** Too much of this can adversely affect performance.

**Important:** The methods in this table are only for use with client scripts.

Table 1383: Client side scripts

<table>
<thead>
<tr>
<th>Script</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>alert(&quot;Hello World&quot;);</code></td>
<td>Will pop up a window with &quot;Hello World&quot; and an 'OK' button.</td>
</tr>
<tr>
<td><code>confirm(&quot;Hello World&quot;);</code></td>
<td>Will pop up a window with &quot;Hello World?&quot; and a 'Ok' and 'Cancel' buttons.</td>
</tr>
<tr>
<td><code>g_form.showFieldMsg(&quot;field_name&quot;, &quot;Hello World&quot;, &quot;error&quot;);</code></td>
<td>Puts &quot;Hello World&quot; in an error message below the specified field.</td>
</tr>
<tr>
<td><code>g_form.hideFieldMsg(&quot;field_name&quot;);</code></td>
<td>Hides an error box that is visible under the specified field.</td>
</tr>
</tbody>
</table>

It is also possible to add other custom messages to your forms if necessary using client scripting.

The text size of info and error messages at the top of the screen is customizable. Two properties control this. If you configured your forms, you may need to add these properties.

Table 1384: Error and alert text size properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>css.outputmsg.info.text.font-size</code></td>
<td>Sets the size for info messages. Default is 11pt.</td>
</tr>
<tr>
<td><code>css.outputmsg.error.text.font-size</code></td>
<td>Sets the size for error messages. Default is 11pt.</td>
</tr>
</tbody>
</table>

**Cost management APIs**

ServiceNow provides APIs for cost management.

See the *ServiceNow Developers site* for API information.

**Glide Server APIs**

ServiceNow provides APIs for the Glide Server.

See the *ServiceNow Developers site* for API information.

**GlideAggregate**

The *GlideAggregate* class is an extension of *GlideRecord* and allows database aggregation (COUNT, SUM, MIN, MAX, AVG) queries to be done. This can be helpful in creating customized reports or in calculations for calculated fields.
For additional information on these APIs, see `GlideAggregate` and `GlideRecord` on the ServiceNow Developers site.

**Note:** This functionality requires a knowledge of JavaScript.

See the ServiceNow Developers site for API information.

**GlideAggregate examples**

GlideAggregate is an extension of GlideRecord and its use is probably best shown through a series of examples.

**Note:** This functionality requires a knowledge of JavaScript.

For additional information, refer to GlideAggregate on the ServiceNow Developers site.

Here is an example that simply gets a count of the number of records in a table:

```javascript
var count = new GlideAggregate('incident');
count.addAggregate('COUNT');
count.query();
var incidents = 0;
if (count.next())
    incidents = count.getAggregate('COUNT');
```

There is no query associated with the above example but if you wanted to get a count of the incidents that were open then you simply add a query just as is done with GlideRecord. Here is an example to get a count of the number of active incidents.

```javascript
var count = new GlideAggregate('incident');
count.addQuery('active', 'true');
count.addAggregate('COUNT');
count.query();
var incidents = 0;
if (count.next())
    incidents = count.getAggregate('COUNT');
```

To get a count of all of the open incidents by category the code is:

```javascript
var count = new GlideAggregate('incident');
count.addQuery('active', 'true');
count.addAggregate('COUNT', 'category');
count.query();
while (count.next()) {
    var category = count.category;
    var categoryCount = count.getAggregate('COUNT', 'category');
    gs.log("The are currently "+ categoryCount + " incidents with a category of "+ category);
}
```

The output is:

*** Script: The are currently 1.0 incidents with a category of Data
*** Script: The are currently 11.0 incidents with a category of Enhancement
*** Script: The are currently 1.0 incidents with a category of Implementation
*** Script: The are currently 197.0 incidents with a category of inquiry
*** Script: The are currently 13.0 incidents with a category of Issue
*** Script: The are currently 1.0 incidents with a category of request
*** Script: The are currently 47.0 incidents with a category of request

Below is an example that is meant to show that you can ask for multiple aggregations. We are asking to see how many times records have been modified and we want the MIN, MAX, and AVG values.

```javascript
var count = new GlideAggregate('incident');
count.addAggregate('MIN', 'sys_mod_count');
```
count.addAggregate('MAX','sys_mod_count');
count.addAggregate('AVG','sys_mod_count');
count.groupBy('category');
count.query();
while(count.next()){
    var min = count.getAggregate('MIN','sys_mod_count');
    var max = count.getAggregate('MAX','sys_mod_count');
    var avg = count.getAggregate('AVG','sys_mod_count');
    var category = count.category.getDisplayValue();
    gs.log(category +" Update counts: MIN = "+ min +" MAX = "+ max +" AVG = "+ avg);
}

The output is:

*** Script: Data Import Update counts: MIN = 4.0 MAX = 21.0 AVG = 9.3333
*** Script: Enhancement Update counts: MIN = 1.0 MAX = 44.0 AVG = 9.6711
*** Script: Implementation Update counts: MIN = 4.0 MAX = 8.0 AVG = 6.0
*** Script: inquiry Update counts: MIN = 0.0 MAX = 60.0 AVG = 5.9715
*** Script: Inquiry / Help Update counts: MIN = 1.0 MAX = 4.0 AVG = 2.0
*** Script: Issue Update counts: MIN = 0.0 MAX = 63.0 AVG = 14.9459
*** Script: Monitor Update counts: MIN = 0.0 MAX = 63.0 AVG = 3.6561
*** Script: request Update counts: MIN = 0.0 MAX = 53.0 AVG = 5.0987

Here is a somewhat more complex example that shows how to compare activity from one month to the next.

```javascript
var agg = new GlideAggregate('incident');
agg.addAggregate('count','category');
agg.orderByAggregate('count','category');
agg.orderBy('category');
agg.addQuery('opened_at','>=','javascript:gs.monthsAgoStart(2)');
agg.addQuery('opened_at','<=','javascript:gs.monthsAgoEnd(2)');
agg.query();
while(agg.next()){
    var category = agg.category;
    var count = agg.getAggregate('count','category');
    var query = agg.getQuery();
    var agg2 = new GlideAggregate('incident');
    agg2.addAggregate('count','category');
    agg2.orderByAggregate('count','category');
    agg2.orderBy('category');
    agg2.addQuery('opened_at','>=','javascript:gs.monthsAgoStart(3)');
    agg2.addQuery('opened_at','<=','javascript:gs.monthsAgoEnd(3)');
    agg2.addEncodedQuery(query);
    agg2.query();
    var last = ";
    while(agg2.next()){last = agg2.getAggregate('count','category');}
    gs.log(category +": Last month:" + count +" Previous Month:" + last);
}
```

The output is:

*** Script: Monitor: Last month:6866.0 Previous Month:4468.0
*** Script: inquiry: Last month:142.0 Previous Month:177.0
*** Script: request: Last month:105.0 Previous Month:26.0
*** Script: Issue: Last month:8.0 Previous Month:7.0
*** Script: Enhancement: Last month:5.0 Previous Month:5.0
*** Script: Implementation: Last month:1.0 Previous Month:0

Here is an example to get distinct count of a field on a group query.

```javascript
var agg = new GlideAggregate('incident');
```
```javascript
agg.addAggregate('count');
agg.addAggregate('count(distinct','category');
agg.addQuery('opened_at', '>=', 'javascript:gs.monthsAgoStart(2)');
agg.addQuery('opened_at', '<=', 'javascript:gs.monthsAgoEnd(2)');
//
agg.groupBy('priority');
agg.query();
while (agg.next()) {
  // Expected count of incidents and count of categories within each priority
  value (group)
gs.info('Incidents in priority ' + agg.priority + ' = ' +
  agg.getAggregate('count') +
  '(' + agg.getAggregate('count(distinct','category') + ' categories');
}
```

The output is:

```text
*** Script: Incidents in priority 1 = 13 (3 categories)
*** Script: Incidents in priority 2 = 10 (5 categories)
*** Script: Incidents in priority 3 = 5 (3 categories)
*** Script: Incidents in priority 4 = 22 (6 categories)
```

**GlideRecord**

GlideRecord is a special Java class (GlideRecord.java) that can be used in JavaScript exactly as if it was a native JavaScript class.

GlideRecord:

- is used for database operations instead of writing SQL queries.
- is an object that contains zero or more records from one table. Another way to say this is that a GlideRecord is an ordered list.

A GlideRecord contains both records (rows) and fields (columns). The field names are the same as the underlying database column names.

For additional information, refer to GlideRecord on the ServiceNow Developers site.

See the ServiceNow Developers site for API information.

**Using GlideRecordSecure**

GlideRecordSecure is a class inherited from GlideRecord that performs the same functions as GlideRecord, and also enforces ACLs.

**Non-Writable Fields**

Be aware that, when using GlideRecordSecure, non-writable fields are set to NULL when trying to write to the database. By default, canCreate() on the column is replaced with canWrite() on the column. If that returns false, the column value is set to NULL.
Checking for NULL Values

If an element cannot be read because an ACL restricts access, a NULL value is created in memory for that record. With GlideRecord, you must explicitly check for any ACLs that might restrict read access to the record. To do so, an if statement such as the following is required to check if the record can be read:

```javascript
if ( !grs.canRead() ) continue;
```

With GlideRecordSecure, you do not need to explicitly check for read access using canRead(). Instead, you can use next() by itself to move to the next record. The following example provides a comparison between GlideRecord and GlideRecordSecure.

```javascript
var count  = 0;
var gr  = new GlideRecord('mytable');
gr. query();
while (gr. next()) {
  if (!gr. canRead()) continue;
  if (!gr. canWrite()) continue;
  if (!gr. val. canRead() || !gr. val. canWrite())
    gr. val = null;
  else
    gr. val = "val-" + gr. id;
  if (gr. update())
    count ++;
}
```

```javascript
var count  = 0;
var grs  = new GlideRecordSecure('mytable');
grs. query();
while (grs. next()) {
  grs. val = "val-" + grs. id;
  if (grs. update())
    count ++;
}
```

Examples

These are two simple examples using GlideRecordSecure.

```javascript
var att  = new GlideRecordSecure ('sys_attachment');
att. get('${sys_attachment.sys_id}');
var sm  = GlideSecurityManager.get();
var checkMe = 'record/sys_attachment/delete';
var canDelete = sm.hasRightsTo(checkMe,att);
gs. log('canDelete: ' + canDelete);
canDelete;
```

```javascript
var grs = new GlideRecordSecure('task_ci');
grs.addQuery();
grs.query();
var count = grs. getRowCount();
if (count > 0 ) {
  var allocation = parseInt(10000/count) / 100;
  while (grs.next()) {
    grs.u_allocation = allocation;
    grs.update();
  }
}
```
GlideSystem

The GlideSystem API provides methods for retrieving information.

The GlideSystem (referred to by the variable name 'gs' in business rules) provides a number of convenient methods to get information about the system, the current logged in user, etc. For example, the method addInfoMessage() permits communication with the user.

```javascript
gs.addInfoMessage('Email address added for notification');
```

Many of the GlideSystem methods facilitate the easy inclusion of dates in query ranges and are most often used in filters and reporting.

For additional information, see GlideSystem on the ServiceNow Developers site.

See the ServiceNow Developers site for API information.

GlideSession

The GlideSession API allows you to find information about the current session.

For additional information, see GlideSession in the ServiceNow Developers site.

See the ServiceNow Developers site for API information.

GlideDateTime

The GlideDateTime class provides methods for performing operations on GlideDateTime objects, such as instantiating GlideDateTime objects or working with glide_date_time fields.

In addition to the instantiation methods described below, a GlideDateTime object can be instantiated from a glide_date_time field using the getGlideObject() method (for example, var gdt = gr.my_datetime_field.getGlideObject();).

Some methods use the Java Virtual Machine time zone when retrieving or modifying a date and time value. Using these methods may result in unexpected behavior. Use equivalent local time and UTC methods whenever possible.

For additional information, refer to GlideDateTime on the ServiceNow Developers site.

See the ServiceNow Developers site for API information.

GlideDate and GlideDateTime examples

The GlideDate and GlideDateTime APIs are used to manipulate date and time values.

**Note:** This functionality requires a knowledge of JavaScript.

For additional information, refer to GlideDate and GlideDateTime on the ServiceNow Developers site.

You can create a GlideDateTime object from a GlideDate object by passing in the GlideDate object as a parameter to the GlideDateTime constructor.

```javascript
var gDate = new GlideDate();
gDate.setValue('2015-01-01');
gs.info(gDate);

var gDT = new GlideDateTime(gDate);
gs.info(gDT);
```

Output:

```
2015-01-01
```
JavaScript tools script includes reference

<table>
<thead>
<tr>
<th>Script Include</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Include class to pull in prototype server if referenced.</td>
</tr>
<tr>
<td>GenericException</td>
<td>Provides generic exception wrapping.</td>
</tr>
<tr>
<td>GSLog</td>
<td>Implements levels of log output to simplify logging and debugging from JavaScript.</td>
</tr>
<tr>
<td>IllegalArgumentException</td>
<td>Provides exception wrapping for a specified illegal argument. This class extends GenericException.</td>
</tr>
<tr>
<td>isJavaObject</td>
<td>Determines if a given value is an instance of a JavaScript object.</td>
</tr>
<tr>
<td>JSONParser</td>
<td>Parses JSON without using JavaScript.</td>
</tr>
<tr>
<td>JSValidator</td>
<td>Provides functions for validating JavaScript code.</td>
</tr>
</tbody>
</table>

GenericException

Provides generic exception wrapping.

Where to Use

Use these methods to define exception handling in your JavaScript code.

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>initialize(errorMessage)</td>
<td>Method called by the Prototype JavaScript Framework during object creation.</td>
</tr>
<tr>
<td>getMessage()</td>
<td>Gets the error message.</td>
</tr>
<tr>
<td>toString()</td>
<td>Converts the error message to a string.</td>
</tr>
</tbody>
</table>

Method Detail

initialize(errorMessage)

Called by the Prototype JavaScript Framework during object creation to initialize a new instance of this class. Provide the parameter value but do not directly call this method.

Input Fields
Parameters:

• **errorMessage** - (string) The error message.

### Output Fields

Returns: void

### Example

```java
var jerr = new GenericException("This is an error");
jerr.getMessage();
```

Gets the error message.

### Input Fields

Parameters:

• None

### Output Fields

Returns: (string) The error message.

### Example

```java
var jerr = new GenericException("This is an error");
gs.print(jerr.getMessage());
```

### Output:

```text
*** Script: This is an error
```

### toString()

Converts the error message to a string.

### Input Fields

Parameters:

• None

### Output Fields

Returns: (string) The error message.

### Example

```java
var jerr = new GenericException("This is an error");
gs.print(jerr.toString());
```

### Output:

```text
*** Script: This is an error
```

### IsJavaObject

Determines if a value is an instance of a Java object rather than a JavaScript primitive, JavaScript object, null, or undefined.
Where to Use

Use in any JavaScript code if you need to test for a Java object instance.

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isJavaObject(val)</td>
<td>Determines if a given value is an instance of a Java object.</td>
</tr>
</tbody>
</table>

Method Detail

isJavaObject(val)
Determines if a given value is an instance of a Java object.

Input Fields

Parameters:
- val - The value to test.

Output Fields

Returns: (boolean) True if the given value is an instance Java object; false, if the value is a JavaScript primitive, JavaScript object, null, or undefined.

Example

```javascript
var jsn = new JSON();
var jsary = [{ name: 'test', value: 1 }, { key: 'test2', value: 2}];
gs.print(isJavaObject(jsn));
```

Output

```
*** Script: false
```

JSValidator

Provides a helper function for validating JavaScript code, by getting any script errors generated by the Glide System. This class extends AbstractAjaxProcessor.

Where to Use

Use this script include whenever you need to validate a script.

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>validate()</td>
<td>Validates the JavaScript code.</td>
</tr>
</tbody>
</table>
Method Detail

validate()
Gets any script errors generated by the Glide System.

Input Fields
Parameters:
• None

Output Fields
Returns: Any script error messages resulting from the validation; if none have occurred, returns null.

Example

```javascript
var jsv = new JSValidator();
gs.print(jsv.validate());
```

Schedule Pages

A schedule page is a record that contains a collection of scripts that allow for custom generation of a calendar or timeline display.

Creation of timeline schedule pages requires understanding how the page/event flow works as well as the ability to write client and server side JavaScript.

Schedule pages form

To access schedule pages, navigate to System Scheduler > Schedules > Schedule Pages.

The form provides the following fields, depending upon the View Type selected:

<table>
<thead>
<tr>
<th>Field</th>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>General name that is used to identity the current schedule page.</td>
</tr>
<tr>
<td>Field</td>
<td>Field Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Schedule type</td>
<td>String</td>
<td>The schedule type is a string that is used to uniquely identify the schedule page via the &quot;sysparm_page_schedule_type&quot; URI parameter. For example, a schedule page could be accessed as follows: /show_schedule_page.do?sysparm_page_schedule_type=gantt_chart&amp;sysparm_timeline_task_id=d530bf907f0000015ce594fd929cf6a4 Alternatively, the schedule page can also be accessed by setting the &quot;sysparm_page_sys_id&quot; URI parameter to that of the unique 32 character hexadecimal system identifier of the schedule page.</td>
</tr>
<tr>
<td>View Type</td>
<td>Choice</td>
<td>Each view type displays different field combinations. There are two options available: • Calendars • Timelines</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>General description that provides additional information about the current schedule page. This field is not necessary.</td>
</tr>
<tr>
<td>Init function name</td>
<td>String</td>
<td>Note: This functionality is only used by Calendar type schedule pages. The init function name specifies the name of the JavaScript function to call inside the Client script function for calendar type schedule pages.</td>
</tr>
</tbody>
</table>
### Field Types and Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>String</td>
<td><strong>Note:</strong> This functionality is only used by Calendar type schedule pages. The HTML field is a scriptable section that is parsed by Jelly and injected into the display page prior to the rest of the calendar. It can be used to pass in variables from the server and define extra fields are necessary.</td>
</tr>
<tr>
<td>Client script</td>
<td>String</td>
<td>The client script is a scriptable section that allows for configuring options of the schedule page display. The API is different depending on the schedule page view type and is discussed below.</td>
</tr>
<tr>
<td>Server AJAX processor</td>
<td>String</td>
<td><strong>Note:</strong> This functionality is only used by Calendar type schedule pages. The Server AJAX processor is specific to calendar type schedule pages that is used to return a set of schedule items and spans to be displayed.</td>
</tr>
</tbody>
</table>

### Timeline Schedule Pages

A Timeline Schedule Page is a specific record that contains configuration information for displaying time based points and spans in a "timeline" like fashion.

The timeline schedule page references a script include that extends from AbstractTimelineSchedulePage to perform dynamic modification to the timeline based on different events and conditions. Both the schedule page and the script include for timeline generation allow for extreme customization and their corresponding application programming interface (API) is documented below.

The following diagram shows the series of events that occur when a timeline schedule page is accessed. Once the timeline has been loaded, all subsequent events, such as events resulting from timeline interaction (e.g. moving a timeline span), follow the same logic flow shown in the gray event box.
Applications that use schedule pages to generate time lines

- Project Management
- Maintenance Schedules
- Group On-Call Rotation
- Field Service Management

Timeline schedule page example

This example demonstrates how to create a timeline schedule page with a script include.

For this example we are going to create an Incident Summary Timeline for a project support manager to visualize all of the new incidents. All new incidents should be displayed as single points where the
priority of the incident is distinguished by a different point icon. Additionally, all closed incidents should be displayed on the timeline in a separate group that shows the duration of the incident before it was closed. Since the Project Manager wants to be able to easily close new items that are resolved without using any form lists, we will handle the vertical move event allowing the new incidents to be dragged into the closed incident group or items within.

**Schedule Page**

Create a new schedule page with the following properties:

- **Name**: Hardware Incidents
- **Schedule type**: incident_timeline
- **View Type**: Timeline
- **Client Script**:

```javascript
// Set our page configuration
glideTimeline.setReadOnly(false);
glideTimeline.showLeftPane(true);
glideTimeline.showLeftPaneAsTree(true);
glideTimeline.showTimelineText(true);
glideTimeline.showDependencyLines(false);
glideTimeline.groupByParent(true);
glideTimeline.setDefaultPointIconClass('milestone');

// We will define what items to display and provide a custom event handler for moving new items to the closed state
glideTimeline.registerEvent('getItems','IncidentTimelineScriptInclude');
glideTimeline.registerEvent('elementMoveY','IncidentTimelineScriptInclude');
```

**Script Include**

Now that the schedule page has been created we need to generate a matching script include for the events that were registered. Create a new script include with the following properties:

- **Name**: IncidentTimelineScriptInclude
- **Active**: Checked
- **Client callable**: Checked
- **Script**:

```javascript
// Class Imports
var IncidentTimelineScriptInclude = Class.create();
IncidentTimelineScriptInclude.prototype =
Object.extendsObject(AbstractTimelineSchedulePage, {

// GET_ITEMS

getItems: function() {
  // Specify the page title
  this.setPageTitle('My Custom Incident Summary Timeline');

  var groupNew = new GlideTimelineItem('new');
groupNew.setLeftLabelText('New Incidents');
groupNew.setImage('../images/icons/all.gifx');
groupNew.setTextBold(true);
this.add(groupNew);
```

© 2017 ServiceNow. All rights reserved. 3970
var groupClosed = new GlideTimelineItem('closed');
groupClosed.setLeftLabelText('Closed Incidents');
groupClosed.setImage('../images/icons/all.gifx');
groupClosed.setTextBold(true);
groupClosed.setIsDroppable(true);

// This allows us to drag an open incident onto the closed group row.
this.add(groupClosed);

// Get all the incidents and let's add only the new/closed ones appropriately
var gr = new GlideRecord('incident');
gr.query();
while(gr.next()) {
    // Only loop through new/closed incidents
    if(gr.incident_state != '1' && gr.incident_state != '7') continue;

    // Ok, we have a new/closed incident. Create the item and the span first.
    var item = new GlideTimelineItem(gr.getTableName(), gr.sys_id);
    var span = item.createTimelineSpan(gr.getTableName(), gr.sys_id);

    // Specific properties for a new incident
    if(gr.incident_state == '1') { // New
        item.setParent(groupNew.getSysId());
        item.setImage('../images/icons/open.gifx');
        span.setTimeSpan(gr.getElement('opened_at').getGlideObject().getNumericValue(),
                         gr.getElement('opened_at').getGlideObject().getNumericValue());

        // We'll show different colors based upon the priorities only for new incidents
        switch(gr.getElement('priority').toString()) {
            case '1': span.setPointIconClass('red_circle'); break;
            case '2': span.setPointIconClass('red_square'); break;
            case '3': span.setPointIconClass('blue_circle'); break;
            case '4': span.setPointIconClass('blue_square'); break;
            case '5': span.setPointIconClass('sepia_circle'); break;
            default: // Otherwise, the default point icon class will be used (Milestone)
                }
    }

    // Specific properties for a closed incident
    else if(gr.incident_state == '7') {
        item.setParent(groupClosed.getSysId());
        item.setImage('../images/icons/closed.gifx');
        span.setTimeSpan(gr.getElement('opened_at').getGlideObject().getNumericValue(),
                         gr.getElement('closed_at').getGlideObject().getNumericValue());

        // Common item properties
        item.setLeftLabelText(gr.short_description);

        // Common span properties
        span.setSpanText(gr.short_description);
        span.setToolTipText('<strong>' + GlideStringUtil.escapeHTML(gr.short_description) + '</strong><br>' +
                             gr.number);
        span.setAllowXMove(false);
        span.setAllowYMove(gr.canWrite() ? true:false);
        span.setAllowYMovePredecessor(false);
        span.setAllowXDragLeft(false);
        span.setAllowXDragRight(false);
    }
}
span.setAllowXDragRight(false);

// Now we add the actual item
this.add(item);
}
},

///////////////////////////////////////////////////////////
ELEMENT_MOVE_Y
///////////////////////////////////////////////////////////

/**
* This is one of the AbstractTimelineSchedulePage event handler methods
* that corresponds to a vertical
* move. The arguments for this function are defined in the API section of
* the event handler methods.
*/
elementMoveY: function(spanId, itemId, newItemId) {

// Get information about the current incident
var gr = new GlideRecord('incident');
gr.addQuery('sys_id', spanId);
gr.query();
if(!gr.next())
    return this.setStatusError('Error', 'Unable to lookup the current
incident.);

// Only allow the new incidents to have their state adjusted.
if(gr.incident_state != '1')
    return this.setStatusError('Error', 'Only new incidents can have
their state adjusted.');

// Get information about the dropped GlideTimelineItem. If it was
dropped in an item on the "New Incidents"
// group let's do nothing. If it was dropped in the "Closed Incidents"
then let's adjust the state automatically.
var grDropped = new GlideRecord('incident');
grDropped.addQuery('sys_id', newItemId);
grDropped.query();
if(!grDropped.next() || grDropped.incident_state == '7') {
    // This means the dropped item was either the 'Closed Incidents'
group (which has no record or sys_id) or an
    // existing incident that is closed. The 'New Incidents' also has no
    sys_id; however, the default behavior for
    // items without a sysId is to be non-droppable. This is why we
    explicitly denoted the 'Closed Incidents' to
    // be marked as "droppable".
    // Return a dialog prompt
    this.setStatusPrompt('Confirm', 'Are you sure you want to close: ' +
    '<div style="margin:10px 0 10px 14px;padding:4px;background-color:#EBEBEB;">'+
    GlideStringUtil.escapeHTML(gr.short_description) +
    '</div></div>',
    'this._elementMoveYHandler_DoClose', // This function is for
    when the OK button is clicked.
    'this._elementMoveYHandler_DoNothing', // This function is
    for when the Cancel button is clicked.
    'this._elementMoveYHandler_DoNothing'); // This function is
    for when the Close button is clicked.
    } },

_elementMoveYHandler_DoClose: function(spanId, itemId, newItemId) {
// Notice that this function takes the same function arguments as the
// original function for which it
// is a custom event handler for.

// Update the database record from 'New' to 'Closed'.
var gr = new GlideRecord('incident');
gr.addQuery('sys_id', spanId);
gr.query();
gr.next();
gr.setValue('incident_state', '7');
gr.update();

// This will re-render the timeline showing the updated item in the
// closed group.
this.setDoReRenderTimeline(true);

// Let's show a success message
this.setStatusSuccess('Success', '<strong>' + gr.short_description + '</strong> was successfully closed.');

// Since the user clicked cancel or close we simply do nothing.
_elementMoveYHandler_DoNothing: function(spanId, itemId, newItemId) {
}};

**Screenshots / Results**

1. **Navigate to:**
   
   http://[YOURINSTANCE]:8080/show_schedule_page.do?
sysparm_page_schedule_type=incident_timeline

   Notice the bold text is the value of the schedule page **Schedule type** field.

2. **The page displays a timeline as specified by the schedule page and script include created.** A link to this page can be created and placed as a module or UI action as necessary.
3. Attempting to move a closed incident anywhere displays the expected error message.

![Error Message Example](image)

Figure 943: Timeline Example Error Moving

4. Moving the incident: I need more memory displays the following confirmation box.

![Confirmation Box Example](image)

Figure 944: Timeline Example Confirm Close

5. Clicking the Cancel button closes the overlay. Clicking the OK button actually updates the incident_state of the record and then displays the following success box.

![Success Message Example](image)

Figure 945: Timeline Example Close Success

6. After clicking OK, it is clear the incident is now listed in the Closed Incidents group.
Figure 946: Timeline Example Incident Updated
Using discovery script includes

Discovery script includes define JavaScript classes that you can use to accomplish Discovery tasks. See the ServiceNow Developers site for API information.

Using GlideRecordUtil to Work with GlideRecords

GlideRecordUtil is a utility class that provides methods that are useful for working with GlideRecords during Discovery.

Getting a GlideRecord Instance

To get a GlideRecord instance for a given configuration item, and of the correct class and table, use the getCIGR(sys_id) method. For example, the following code gets the GlideRecord of a CI with the sys_id of 2dfd7c8437201000deeabfc8bcbe5d56:

```javascript
var gr = new GlideRecordUtil().getCIGR("2dfd7c8437201000deeabfc8bcbe5d56");
```

To retrieve any hierarchical table without knowing its class type, use the getGR(base_table, sys_id) method. For instance, if you need to get a GlideRecord for a computer class CI, you may not know if it is a computer class or more specifically a Windows server class or Linux server class. Using this method guarantees that you have a GlideRecord with the correct class. This is important because different classes have different attributes—in this case, a Windows server has attributes different from those of a Linux server. You must get a GlideRecord in the correct class or attributes may be missing. The following is a typical example of how to do this:

```javascript
var gr = new GlideRecordUtil().getGR("cmdb_ci_computer", "2dfd7c8437201000deeabfc8bcbe5d56");
```

Getting All the Fields In a GlideRecord

The getFields(gr) method returns a JavaScript object, such as a hash map, of all the fields or attributes that exist in a given GlideRecord.

```javascript
var gr = new GlideRecordUtil().getGR("cmdb_ci_computer", "2dfd7c8437201000deeabfc8bcbe5d56");
var fields = new GlideRecordUtil().getFields(gr);
gs.log(fields.join(" ")); // List all the fields that are in a computer CI
```

Populating GlideRecord Object Fields

The populateFromGR(hashmap, gr, ignore) method allows you to take a GlideRecord object and populate its fields and values into a JavaScript object. The third argument (ignore) is an optional JavaScript object that allows you to exclude certain fields. For example, you may not care about sys_created_by or sys_updated_by fields in a GlideRecord.

```javascript
var objectToPopulate = { };
var gr = new GlideRecordUtil().getGR("cmdb_ci_computer", "2dfd7c8437201000deeabfc8bcbe5d56");
var ignore = {"sys_created_on": true, "sys_updated_by": true};
new GlideRecordUtil().populateFromGR(objectToPopulate, gr, ignore);
```
// Now the objectToPopulate contains field/value pairs from the computer GlideRecord

The `mergeToGR(hashmap, gr, ignore)` method allows you to populate a GlideRecord with a field/value-paired object. The ignore argument stops specified fields from being updated. The following code example updates a computer record’s name and os fields, but does not update the sys_created_by field:

```javascript
var gr = new GlideRecordUtil().getGR("cmdb_ci_computer", "2dfd7c8437201000deeabfc8bcbef5d56");
var obj = {"name": "xyz", "os": "windows 2000", "sys_created_by": "aleck.lin"};
var ignore = {"sys_created_by": true};
new GlideRecordUtil().mergeToGR(obj, gr, ignore);
gr.update();
```

**Getting Table Hierarchies**

The `getTables(table)` method returns a list of table hierarchies, as shown in the following example:

```javascript
var tables = new GlideRecordUtil().getTables("cmdb_ci_linux_server");
gs.log(tables.join(",");
// The result would be "cmdb_ci, cmdb_ci_computer, cmdb_ci_server, cmdb_ci_linux_server".
```

**Using DiscoveryException and AutomationException**

When writing Discovery sensors and sensor-related scripts, you may want to use DiscoveryException or AutomationException to indicate that an exception has come from Discovery.

The `DiscoveryException` script include extends `AutomationException`, which extends the `GenericException` class. The following example uses `DiscoveryException` to throw an exception:

```javascript
function foo() {
    if(//condition matches) throw new DiscoveryException("The message", "The cause");
}
```

The first argument takes the message of the exception and the second argument (optional) takes the cause of the exception. You may also want to catch the exception and log it as shown in the example below:

```javascript
try {
    foo();
}
catch(e) {
    if(e instanceof DiscoveryException)
        gs.log("A DiscoveryException occurred. It is " + e.getMessage() + " caused by " + e.getCause());
}
```

The above example also applies for `AutomationException`. `DiscoveryException` is typically used to provide exception processing specifically for Discovery, while `AutomationException` is used for exception processing that applies to both Orchestration and Discovery.

**vCenter API User Privileges**

To determine the user privileges required by the VMware activity that logs into vCenter:
Determining Privileges

1. Navigate to the VMware API documentation.
2. In the vCenter API page, select **All Types** in the left navigation menu.

3. Click **U-Z** in the API index that appears.

4. Select **VirtualMachine** in the index.
   
   Information appears about a VirtualMachine object in the API.

5. Select **Local Methods** from the navigation menu at the top of the page.
6. Click the following methods to see the required vCenter login privileges:
   - CloneVM_Task: No privileges are required.

Methods

Methods inherited from ManagedEntity

Destroy_Task, Reload, Rename_Task

Methods inherited from ExtensibleManagedObject

setCustomValue
XMLDocument script object

A JavaScript object wrapper for parsing and extracting XML data from an XML document (String).

Use this Javascript class to instantiate an object from an XML string, usually a return value from a Web Service invocation, or the XML payload of ECC Queue. Using the XMLDocument object in a Javascript business rule lets you query values from the XML elements and attributes directly.

Programming Interface

Constructor

The constructor of a script object creates a new instance of the object to be used.

```javascript
var xmlString = "<test>
  <one>
    <two att="xxx">abcd1234</two>
    <three boo="yah" att="yyy">1234abcd</three>
    <two>another</two>
  </one>
  <number>1234</number>
</test>";
var xmldoc = new XMLDocument(xmlString);
```

To perform XML parsing of an XML string that is name space qualified, specify "true" for the second argument for the XMLDocument constructor. The following is an example of parsing and XML string that contains name space qualification of its elements.

```javascript
var xmlString = "<bk:book xmlns:bk='urn:loc.gov:books'
  <bk:title>Cheaper by the Dozen</bk:title>
  <isbn:number>1568491379</isbn:number>
</bk:book>";
var xmldoc = new XMLDocument(xmlString, true); // XML document is name space aware
```

Locating Nodes and Elements

Now that we have the XMLDocument object, we can call the following APIs to locate nodes and elements of the XML document, as well as extract text from it directly. The query syntax for locating nodes and attributes is based on XPath.

**Note:** XML parsing with this object is not namespace aware, this means that if you are locating a node name with namespace in it eg. "<names:myelement> ...", the XPath search string would be "/names/myelement"

The following are examples of locating a node by its XPath and getting the text value out of the resulting node.

```javascript
var str = xmldoc.getNodeText("//two"); // returns the first occurence of the node
// str == "abcd1234"
```
The following examples locates the node by XPath and uses the API on node and element to get attributes and value.

```javascript
var node = xmldoc.getNode("/test/one/two");
// node.getNodeName() == "two"
// node.getNodeType() == "1" // 1 == ELEMENT_NODE
// node.getLastChild().getType() == "3" // 3 == TEXT_NODE
// node.getLastChild().getNodeValue() == "abcd1234"
```

Or you can use the following convenience functions to get the node name and type

```javascript
str = xmldoc.getNodeName("//three");
// str == "three"

str = xmldoc.getNodeType("//three");
// str == "1"
```

You can also get a list of nodes that you can iterate or access directly by position

```javascript
var nodelist = xmldoc.getNodes("//one/*"); // two, three, and two
// nodelist.getLength() == "3"
// nodelist.item(0).getNodeName() == "two"
// nodelist.item(1).getNodeName() == "three"
```

The following is an example of parsing an XML string that is qualified with name spaces.

```javascript
var xmlString = "<bk:book xmlns:bk='urn:loc.gov:books'
xmlns:isbn='urn:ISBN:0-395-36341-6'>" +
"<bk:title>Cheaper by the Dozen</bk:title>" +
"<isbn:number>1568491379</isbn:number>" +
"</bk:book>";

var xmldoc = new XMLDocument(xmlString, true);
var str = xmldoc.getNodeText("//bk:title"); // returns the first occurrence of the node
gs.log(str);

str = xmldoc.getNodeText("//bk:book/*");
gs.log(str);

str = xmldoc.getNodeInt("//isbn:number");
gs.log(str);
```

**Getting attribute values**

An attribute is just an extension of a node and so it has all the same APIs.

The following example shows how to query for a node to get its attribute by position

```javascript
node = xmldoc.getNode("//two");
```
XPath also has a query syntax for locating the attribute node directly, for example

```
str = xmldoc.getAttribute("//two", "att");
// str == "xxx"
```

```
str = xmldoc.getNodeText("//*[@att="yyy"]");
// str == "1234abcd"
```

```
str = xmldoc.getNode("//@boo").getNodeValue();
// str == "yah"
```

Creating new elements

An XML element can be created at any level of the XML document once it has been created. Use the `setCurrent` function to position where the new element will be created as a child element, and use the `createElement` function to create the element.

```
var xmlString = "<test>"
  "  <one>
    "<two att="xxx">abcd1234</two>
    "<three boo="yah" att="yyy">1234abcd</three>
    "<two>another</two>
  "</one>
  "<number>1234</number>"
"</test>";
```

```
var xmldoc = new XMLDocument(xmlString);
xmldoc.createElement("new", "test"); // creates the new element at the document element level if setCurrent is never called
xmldoc.createElement("new2"); // calling without a value will create a new element by itself
```

```
var el = xmldoc.createElement("new3");
xmldoc.setCurrent(el); // this is now the parent of any new elements created subsequently using createElement()
xmldoc.createElement("newChild", "test");
```

The resulting XML document looks like this

```
<test>
  <one>
    <two att="xxx">abcd1234</two>
    <three boo="yah" att="yyy">1234abcd</three>
    <two>another</two>
  </one>
  <number>1234</number>
  <new test="new" new2/> <new3>
    <newChild>test</newChild>
  </new3>
</test>
```

XMLHelper

The XML helper script include makes it extremely easy to parse XML in ServiceNow JavaScript. The script include will convert your XML document into a JavaScript object.
The following changes were made to the XML helper script include:

- The `toObject()` method now returns an object whose properties are all JavaScript objects. This method now works properly whether the supplied parameter is an XML document or an XML string.
- The methods `toXMLDoc()` and `toXMLStr()` are now available. These methods are the inverse of the existing `toObject()` method.
- The `toObject()` method has been extended to take an optional parameter of the XML input to convert as an alternative to the (still present) mechanism of specifying the XML input in the constructor.

**Example**

The following example takes an example XML document and then converts it into a JavaScript object. It then takes the outputted JavaScript object and uses a recursive function to output all members of the object. The recursive function is useful and reusable if you have any questions about how a particular XML document will be structured after being converted to a JavaScript object.

**Script**

```javascript
var xmlString = "<company>
  <employee>
    <id>10</id>
    <firstname>Tom</firstname>
    <lastname>Cruise</lastname>
    <test>test1</test>
    <test>test3</test>
  </employee>
  <employee>
    <id>20</id>
    <firstname>Paul</firstname>
    <lastname>Enderson</lastname>
    <test>test6</test>
    <test>test5</test>
  </employee>
  <employee>
    <id>30</id>
    <firstname>Paul</firstname>
    <lastname>Bush</lastname>
    <test>test2</test>
    <test>test4</test>
  </employee>
</company>";

var helper = new XMLHelper(xmlString);
var obj = helper.toObject();

logObj(obj, "**")
function logObj(obj, sep){
  for(x in obj){
    if(typeof obj[x]!="function"){
      gs.log(sep + x +": " + obj[x]);
      logObj(obj[x], sep +"*");}
  }
}
```

**Output**

```
*** Script: *employee:: [object Object], [object Object], [object Object]
*** Script: **2:: [object Object]
*** Script: ***id:: 30
*** Script: ****test:: test2, test4
*** Script: *****0:: test2
*** Script: ****1:: test4
*** Script: ***firstname:: Paul
*** Script: ****lastname:: Bush
*** Script: **0:: [object Object]
*** Script: ***id:: 10
*** Script: ****test:: test1, test3
*** Script: *****0:: test1
*** Script: ****1:: test3
*** Script: ***firstname:: Tom
*** Script: ****lastname:: Cruise
*** Script: **1:: [object Object]
*** Script: ***id:: 20
```
Useful scripts

Scripts that provide useful functionality not included in the core system.

Get a user object

In a business rule or other server script, the `gs.getUser()` method returns a user object. The user object is an internal representation of the currently logged in user and provides information about the user and various utility functions.

See the ServiceNow Developers site for API information.

For a list and description of the available scoped methods for the user object, see `GlideUser()` on the ServiceNow Developers site.

To get user information for a particular user:

1. Retrieve the current user.
   
   ```javascript
   var myUserObject = gs.getUser()
   ```

2. Use the `getUserByID` method to fetch a different user using the `user_name` field or `sys_id` on the target record.
   For example:

   ```javascript
   var ourUser = gs.getUser();
   gs.print(ourUser.getFirstName()); //print the first name of the user you are currently logged in as
   ourUser = ourUser.getUserByID('-abel.tuter'); //fetch a different user, using the user_name field or sys_id on the target user record.
   gs.print(ourUser.getFirstName()); //first name of the user you fetched above
   gs.print(ourUser.isMemberOf('-Capacity Mgmt'));
   ```

Accessing the workflow scratchpad from business rules

A catalog item has been requested, the attached workflow contains a run script activity that populates a value in the scratchpad. From a business rule running on the requested item, we want to retrieve or set scratchpad values.

Prerequisites

Role required: admin

Name: Access Workflow Scratchpad from Business Rules

Type: Business Rule

Table: sc_req_item (Requested Item)
**Description:** A catalog item has been requested, the attached workflow contains a run script activity that populates a value in the scratchpad. From a business rule running on the requested item, we want to retrieve or set scratchpad values.

**Parameters:** n/a

**Script:**

```javascript
//the run script activity sets a value in the scratchpad
workflow.scratchpad.important_msg = "scratch me";

//get the workflow script include helper
var workflow = new Workflow();

//get the requested items workflow context
//this will get all contexts so you'll need to get the proper one if you have multiple workflows for a record
var context = workflow.getContexts(current);
//make sure we have a valid context
if (context.next()) {
  //get a value from the scratchpad
  var msg = context.scratchpad.important_msg;
  //msg now equals "scratch me", that was set in the run script activity

  //add or modify a scratchpad value
  context.scratchpad.status = "completed";
  //we need to save the context record to save the scratchpad
  context.update();
}
```

Add a field to the service catalog checkout

This is an example of adding a **Company** field to the checkout below the **Requested for** field.

![Requested for field](image)

**Figure 947: Requested for field**

This field will then passing the value of that field to the **Company** field of the Service Catalog Request.

This example makes the following assumptions.

- This example is for an instance using two-step checkout. If two-step checkout is not enabled, enable it before beginning.
- This example populates the **Company** field on the Service Catalog Request form. If the field does not appear on the form, configure the form before beginning.

To add a field to the Service Catalog Checkout:

1. Go to **System UI > UI Macros** and select **servicecatalog_cart_template**.
2. Find where there are hidden variables declared and add the following line:

```html
<input type="HIDDEN" name="cart_id" id="cart_id" value="${sc_cart.sys_id}" />
```

3. Find the following code, which generates the **Requested For** code:

```html
<tr class="header">
    <td width="30%">
        ${gs.getMessage('Requested for')}:
    </td>
    <td width="70%">
        <label for="requestor_location">${gs.getMessage('Deliver to')}:</label>
    </td>
</tr>
<tr><td>$[SP]</td></tr>
<tr><td valign="top">
    <j2:if test="$[jvar_can_delta_rf == false]">
        $[sc_cart.requested_for.getDisplayValue()]
    </j2:if>
    <j2:if test="$[jvar_can_delta_rf != false]">
        <g2:catalog_requested_for />
    </j2:if>
</td>
<td>
    <textarea id="requestor_location" style="width:100%" rows="4"
        name="requestor_location" wrap="soft"
        onChange="catDeliveryAddress('${sc_cart.sys_id}',
            'requestor_location');">
        $[sc_cart.delivery_address]
    </textarea>
</td>
<tr><td>$[SP]</td></tr>
</tr>
<tr><td colspan="2">Company</td></tr>
<tr><td>$[SP]</td></tr>
<tr><td colspan="2">
    <g2:ui_reference name="core_company" table="core_company"
        onchange="setCartValue()"/>
</td></tr>
<tr><td>$[SP]</td></tr>
</tr>
```

4. Add the following code afterwards:

```html
<tr class="header">
    <td colspan="2">Company</td>
</tr>
<tr><td>$[SP]</td></tr>
<tr><td colspan="2">
    <g2:ui_reference name="core_company" table="core_company"
        onchange="setCartValue()"/>
</td></tr>
<tr><td>$[SP]</td></tr>
</tr>
```

- Note that 'ui_reference' is another macro that defines a reference field. There are several macros for different field types. You can see examples of these field types under **System UI -> UI Macros**. These macros start with 'ui_'. For this example, the reference field created is named **core_company**.
5. Now navigate to System UI > UI Pages and select the servicecatalog_checkout_one UI Page. Add this script to the Client script field:

```javascript
function setCartValue() {
  var newField = gel('core_company');
  var myCart = gel('cart_id');
  var cart_item = new GlideRecord('sc_cart_item');
  cart_item.addQuery('cart', myCart.value);
  cart_item.query();
  if(cart_item.next()) {
    cart_item.hints = "<hints><entry key='sysparm_processing_hint' value='setfield:request.company=' + newField.value + "/'></hints>>";
    cart_item.update();
  }
}
```

For this example, the reference field was called core_company, and the field being populated on the request is company. If different fields are used:

- Find this line: var company = gel('core_company'); and replace core_company with the name of the field in the checkout.
- In the line that starts with 'cart_item.hints' replace 'request.company' with the name of the field to be populated on the request ticket where 'request' is the request being generated and 'company' is the name of the field.

Now, when an item is ordered, the company field appears on the Catalog form:
Add role to every user

Adds a role to every user.

Prerequisites

**Note:** This functionality requires a knowledge of JavaScript.

Role required: admin

In this sample, the role being added is Self Service. To add a different role, simply substitute the desired role for self_service.

**Name:** Add Role to Every User  
**Type:** Client Script, Background Script  
**Table:** sys_user  
**Description:** Adds a role to every user. In this sample, the role being added is **Self Service**. To add a different role, simply substitute the desired role for **self_service**.
Assign a catalog item to a group based on a delivery plan task

This assignment rule assigns a service catalog item to the database group if it uses a delivery plan that has a catalog task assigned to the desktop group.

**Prerequisites**

Role required: admin

---

**Caution:** The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

**Name:** Assign Catalog Item to Group Based on Delivery Plan Task

**Type:** Assignment Rule

**Table:**

**Description:** This assignment rule assigns a service catalog item to the database group if it uses a delivery plan that has a catalog task assigned to the desktop group.

**Parameters:**

**Script:**

```javascript
//Return catalog items that have no group but do have a delivery plan assigned
var ri = new GlideRecord("sc_cat_item");
ri.addQuery("group", ",\nnull);
ri.addQuery("delivery_plan", ",\n!=", null);
ri.query();
while(ri.next()) {
    gs.log("Found an item");
    //Return tasks that point to the same delivery plan as the above item
    var dptask = new GlideRecord("sc_cat_item_delivery_task");
dptask.addQuery("delivery_plan", ",\n!=", ri.delivery_plan);
dptask.query();
    while(dptask.next()) {
        gs.log("Found a task");
        var gp = dptask.group.getDisplayValue();
gs.log(gp);
        //If the task is assigned to desktop, assign the item's group to desktop
        if (dptask.group.getDisplayValue() == "Desktop") {
            ri.group.setDisplayValue("Desktop");
gs.log("updating " + ri.getDisplayValue());
            ri.update();
        }
    }
}
```
Change form color on state change

Changes color of a form field of the form on state change. The script can easily be changed to adjust any property of any object on the page accessible via the HTML DOM.

**Caution:** The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

Name: Change Form Color on State Change
Type: Client Script
Table:
Description: Changes color of a form field of the form on state change. The script can easily be changed to adjust any property of any object on the page accessible via the HTML DOM.
Parameters:
Script:

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    var elementID = gel("incident.priority");
    switch (newValue) {
        case "1":
            elementID.style.backgroundColor = "red";
            break;
        case "2":
            elementID.style.backgroundColor = "tomato";
            break;
        case "3":
            elementID.style.backgroundColor = "orange";
            break;
        case "4":
            elementID.style.backgroundColor = "yellow";
            break;
        case "5":
            elementID.style.backgroundColor = "green";
            break;
        default:
            elementID.style.backgroundColor = "white";
            break;
    }
}
```

Create a UI routing action

This solution enables you to create a record with the service desk without knowing whether it is an incident or request item; the service desk can then route the record to the appropriate table.

**Note:** Functionality described here requires the Admin role.

To create a UI routing action:

1. Create a new table that extends the task table (for example, New Call).
2. Create a module to create a new New Call record.
3. Create any fields that you want on the New Call table.

   The only fields you need are those fields necessary to determine whether the new call should route to an Incident or a Request Item. Ensure that the form contains any fields that you want to pass to the Incident or Request Item. In this example, the following are created on the form:
   - Requested for (reference)
   - Location (reference)
   - Call type (choice with two values--Incident and Request)
   - Request Item (reference to the sc_cat_item Item table)

4. Add some UI policies to set a couple of fields to mandatory and hide the Request item field based on the Call type selection.
5. Remove unnecessary buttons and functionality from the form.

6. Create a new UI Action button. This button redirects the user to either an incident or a request. It also creates the incident record and copies values to the incident and the Request Item form.

```javascript
var reqItem = current.u_item;
var requestedFor = current.u_requested_for;
var location = current.location;

if(current.u_incident_request == 'Incident'){
    //Create a new incident record and redirect to the new incident
    var rec = new GlideRecord('incident');
    rec.initialize();
    rec.caller_id = requestedFor;
    rec.location = location;
    rec.insert();
    action.setRedirectURL(rec);
}

if(current.u_incident_request == 'Request'){
    //Build the url and route the user to the request item
    var url = '';
    if(current.u_item.sys_class_name == 'sc_cat_item_guide'){
        url = 'com.glideapp.servicecatalog_cat_item_guide_view.do?' +
              'sysparm_initial=true&sysparm_guide=' +
              reqItem + '&sysparm_user=' + requestedFor + '&sysparm_location=' +
              location;
    } else{
        url = 'com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=' +
              reqItem + '&sysparm_user=' +
              requestedFor + '&sysparm_location=' + location;
    }
    action.setRedirectURL(url);
}
```

7. The Route button in the preceding example passes the Requested for and Location values in the URL to the Request Item form. Ensure that you have variables called requested_for and location on your item, record producer, or order guide that map these values using the following client script. There is a limit as to how much information you can pass, as the URL has a restricted length. It is generally not a best practice to send information from long text fields using this method.

```javascript
function onLoad() {
    var url = document.location.toString();
    var userKey = 'sysparm_user=';
    var locKey = 'sysparm_location=';
    var userPosition = url.indexOf(userKey);
    var locPosition = url.indexOf(locKey);
    if (userPosition != -1) {
        var user = url.substr(userPosition+userKey.length, 32);
        g_form.setValue('requested_for',user);
    }
    if (locPosition != -1) {
        var loc = url.substr(locPosition+locKey.length, 32);
        g_form.setValue('location',loc);
    }
}
```

Custom approval UI macro

This section describes how to create a custom approval UI macro.
**Caution:** The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

**Script Name:** Custom Approval UI Macro  
**Type:** UI Macro  
**Description:** Here is an option to get more detail out of the My Approvals view of an Execution Plan. This can be done by creating a new UI macro. Navigate to System UI and click UI Macros. First, you will need to rename the existing approval_summarizer_sc_task to something like approval_summarizer_sc_task_old and deactivate it. Then you will need to create a new one using the same name (approval_summarizer_sc_task). The name should basically tell you what the macro does and what it applies to. In this case, we’re replacing an existing one so we decided to re-use the existing name.

Then you should copy the xml script at the bottom of this article into the xml code window in the new UI macro. This is great way to give some detail to an approver when you are doing line item approvals via approval tasks within the Service Catalog Execution Plans.
The old way

This is the view you see in My Approvals when using an approval task via the old method.

Figure 949: My Approvals (old way)

Notice there is not much detail telling the approver what they are actually approving. You can see the short description of the task but not much around what the item is.

The new way

This is the view you will see if you use the xml script below in place of the OOB (out-of-box) UI macro.
Using this method you can see details much like the request approval. You have a link into the item ordered, a short description (which contains the ability to expand the variables from the item), price, quantity and the total price. This helps the approver in that it shows more detail. They can now see what they are actually approving.

Script:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<j:jelly trim="true" xmlns:j="jelly:core" xmlns:g="glide"
xmlns:j2="null" xmlns:g2="null">
<tr>
  <td class="label_left" width="100%">
    <label style="margin-left: 10px">
      ${gs.getMessage('Summary of Item being approved')}:<input style="visibility: hidden" NAME="make_spacing_ok"></input>
    </label>
  </td>
</tr>
<g:evaluate var="jvar_ni" expression="
  var sc_task = ${ref}.sysapproval;
  var sc_req_labels = new GlideRecord('sc_req_item');
  sc_req_labels.initialize();
  var sc_req_item = new GlideRecord('sc_req_item');
  sc_req_item.addQuery('request_item', sc_task.request_item.sys_id);
  sc_req_item.query();
" />
<tr>
  <td width="100%">
    <table width="100%">
      <tr>
        <td class="label_left" width="150px">
          <label style="margin-left: 10px">
            ${sc_task.request_item.request.opened_by.sys_meta.label}:
          </label>
        </td>
      </tr>
      <tr>
        <td class="label_left" width="150px">
          ${sc_task.request_item.request.opened_by.sys_meta.label}:
        </td>
        <td>
          $25.00
        </td>
        <td>
          2
        </td>
        <td>
          $50.00
        </td>
      </tr>
    </table>
  </td>
</tr>
</j:jelly>
```
<td>${sc_task.request_item.request.opened_by.getDisplayValue()}</td>

<tr>
<td class="label_left" width="150px">
<label style="margin-left: 10px">${sc_task.request_item.request.requested_for.sys_meta.label}:</label>
</td>
<td>${sc_task.request_item.request.requested_for.getDisplayValue()}</td>
</tr>
<tr>
<td class="label_left" width="150px">
<label style="margin-left: 10px">${gs.getMessage('Total')}:</label>
</td>
<td><g:currency_format value="${sc_task.request_item.request.price}" /></td>
</tr>
</table>

<j:set var="jvar_line_num" value="0" />
<tr>
<td width="100%">
<table width="100%">
<tr class="header">
<td colspan="2">${sc_req_labels.number.sys_meta.label}</td>
<td>${sc_req_labels.description.sys_meta.label}</td>
<td>${sc_req_labels.price.sys_meta.label}</td>
<td>${sc_req_labels.quantity.sys_meta.label}</td>
<td>${gs.getMessage('Total')}</td>
</tr>
<j:set var="jvar_item_price" value="${sc_task.request_item.price * sc_task.request_item.quantity}" />
<j:set var="jvar_overall_total" value="${jvar_overall_total + jvar_item_price}" />
<j:set var="jvar_line_color" value="odd" />
<j:set var="jvar_line_num" value="${jvar_line_num + 1}" />
<j:if test="${jvar_line_num % 2 == 0}">
<j:set var="jvar_line_color" value="even" />
</j:if>
<g:evaluate var="ni" expression="var smart_description = sc_task.request_item.cat_item.short_description;" />
if (smart_description == null || smart_description == '')
    smart_description =
sc_task.request_item.cat_item.name;
"/>
    <tr class="${jvar_line_color}">
        <td valign="top">
            <g2:evaluate var="jvar_pop_target" expression="${[ref].getRecordClassName()}">
                <a class="linked" target="gsft_main" href="sc_req_item.do?sys_id=${sc_task.request_item.sys_id}" onmousemove="popListDiv(event, 'sc_req_item', '${sc_task.request_item.sys_id}', '${sysparm_view}')" onmouseout="lockPopup(event)" class="clsshort"></a>
            </td>

            <td valign="top">
                <a class="linked" target="gsft_main" href="sc_req_item.do?sys_id=${sc_task.request_item.sys_id}">
                    ${sc_task.request_item.number}
                </a>
            </td>

            <td valign="top" width="50%">
                <g:call function="variable_summary_approval.xml" question_name="${sc_task.request_item.sys_id}" question_help_tag="${smart_description}" sc_req_item="${sc_task.request_item.sys_id}" help_class="${jvar_line_color}" />
            </td>

            <td valign="top">
                <g:currency_format value="${sc_task.request_item.price}/></td>
            </td>

            <td valign="top"> ${sc_task.request_item.quantity}</td>

            <td valign="top">
                <g:currency_format value="${jvar_item_price}="/></td>
        </tr>
    </table>
</td>
</tr>
</j:jelly>

Display field messages

Rather than use JavaScript alert(), for a cleaner look, you can display an error on the form itself. The methods `showFieldMsg()` and `hideFieldMsg()` can be used to display a message just below the field itself.

`showFieldMsg` and `hideFieldMsg` are methods that can be used with the `g_form` object.

These methods are used to make changes to the form view of records (Incident, Problem, and Change forms). These methods may also be available in other client scripts, but must be tested to determine whether they will work as expected.

When a field message is displayed on a form on load, the form will scroll to ensure that the field message is visible, ensuring that users will not miss a field message because it was off the screen.

A global property (glide.ui.scroll_to_message_field) is now available that controls automatic message scrolling when the form field is offscreen (scrolls the form to the control or field).
### Table 1386: Method Detail

<table>
<thead>
<tr>
<th>Method Detail</th>
<th>Parameters</th>
<th>Example</th>
</tr>
</thead>
</table>
| `showFieldMsg(input, message, type, [scrollform])` | - input - name of the field or control  
- message - message you would like to display  
- type - either 'info' or 'error', defaults to info if not supplied  
- scroll form - (optional) Set scrollForm to false to prevent scrolling to the field message offscreen. | **Error Message**  
g_form.showFieldMsg('impact','Low impact not allowed with High priority','error');  
| **Informational Message**  
g_form.showFieldMsg('impact','Low impact response time can be one week','info');  
//or this defaults to info type  
//g_form.showFieldMsg('impact','Low impact response time can be one week'); |
| `hideFieldMsg(input)` | - input - name of the field or control  
- clearAll - (optional) boolean parameter indicating whether to clear all messages. If true, all messages for the field will be cleared; if false or empty, only the last message will be removed | **Removing a Message**  
//this will clear the last message printed to the field  
g_form.hideFieldMsg('impact'); |

### Legacy support

The `showErrorBox()` and `hideErrorBox()` are still available but simply call the new methods with type of error. It is recommended that you use the new methods.

### GSLog

GSLog is a script include which makes logging and debugging from script easier by implementing levels of log output, selectable by per-caller identified sys_properties values.
Log level

Logs can be at the level of Debug, Info, Notice, Warning, Err, or Crit (after BSD syslog.h and followers). The default logging level would normally be Notice, so levels should be chosen accordingly.

Where to use

Use for any server-side script where you want to implement event logging.

Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Input Fields</th>
<th>Output Fields</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>initialize(traceProperty, caller)</td>
<td>Called by the Prototype JavaScript Framework during object creation to initialize a new instance of this class. Provide the input parameters, but do not call this method directly.</td>
<td>Parameters: - traceProperty - A system property that contains a value indicating the level at or above which messages will be written to the log. - caller - The name of the script calling the logger.</td>
<td>Returns: void</td>
<td>var gl = new GSLog(&quot;com.snc.sla.tasksla.log&quot;, &quot;TaskSLA&quot;);</td>
</tr>
</tbody>
</table>
| logDebug(msg)  | Logs debug events.            | Parameters: msg - The message to write to the log.                          | Returns: void | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");
<p>|                |                              |                                                                              |               | gl.logDebug(&quot;This is a debug message&quot;);                               | Output: *** Script [TaskSLA]: This is a debug message |
| logInfo(msg)   | Logs information events.      | Parameters: msg - The message to write to the log.                          | Returns: void | var gl = new GSLog(&quot;com.snc.sla.tasksla.log&quot;, &quot;TaskSLA&quot;);             |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Input Fields</th>
<th>Output Fields</th>
<th>Example</th>
</tr>
</thead>
</table>
| logNotice(msg) | Logs notice events.    | Parameters: msg  
- The message to write to the log. | Returns: void | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");  
gl.logNotice("This is a notice");  

Output:  
*** Script [TaskSLA]:  
This is a notice |
| logWarning(msg) | Logs warning events. | Parameters: msg  
- The message to write to the log. | Returns: void | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");  
gl.logWarning("This is a warning message");  

Output:  
*** Script [TaskSLA]:  
This is a warning message |
| logErr(msg)    | Logs error events.    | Parameters: msg  
- The message to write to the log. | Returns: void | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");  
gl.logErr("This is an error message");  

Output:  
*** Script [TaskSLA]:  
This is an error message |
| logCrit(msg)  | Logs critical events. | Parameters: msg  
- The message to write to the log. | Returns: void | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");  
gl.logCrit("This is a critical message");  

Output:  
*** Script [TaskSLA]:  
This is a critical message |
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Input Fields</th>
<th>Output Fields</th>
<th>Example</th>
</tr>
</thead>
</table>
| logAlert   | Logs alert events.           | Parameters: msg - The message to write to the log.                            | Returns: void     | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");
gl.logAlert("This is a critical message");                                                                                                                                                                                                                                                                                                                                                                                                                      |
| logEmerg   | Logs emergency events.       | Parameters: msg - The message to write to the log.                            | Returns: void     | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");
gl.logEmerg("This is an emergency message");                                                                                                                                                                                                                                                                                                                                                                                                                       |
| log        | Logs a message at the specified level. | Parameters: • level - The log level. • msg - The message to write to the log. | Returns: void     | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");
gl.log("debug", "debug message");                                                                                                                                                                                                                                                                                                                                                                                                                        |
| setLevel   | Sets the log level.          | Parameters: level - The log level.                                           | Returns: void     | var gl = new GSLog("com.snc.sla.tasksla.log", "TaskSLA");

© 2017 ServiceNow. All rights reserved.
**Modify a GlideDateTime field value**

This example demonstrates how to modify a GlideDateTime field value using a script.

**Name:** Modify a GlideDateTime Field Value

**Type:** A server side script that accesses a GlideDateTime field.

**Table:** N/A

**Description:** Given a GlideDateTime field or script object, show a variety of ways to easily modify value. The same concept also applies to the GlideDate object.

**Parameters:** N/A

**Script:**

```java
//You first need a GlideDateTime object
//this can be from instantiating a new object "var gdt = new GlideDateTime()"
//or getting the object from a GlideDateTime field
//getting the field value (for example: var gdt = current.start_date) only returns the string value, not the object
//to get the object use var gdt = current.start_date.getGlideObject();
//now gdt is a GlideDateTime object
var gdt = current.start_date.getGlideObject();

//All methods can use negative values to subtract intervals

//add 1 hour (60 mins * 60 secs)
gdt.addSeconds(3600);

//add 1 day
gdt.addDaysLocalTime(1);
```

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Input Fields</th>
<th>Output Fields</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getLevel(level)</code></td>
<td>Gets the log level.</td>
<td>Parameters: level - (optional) The log level.</td>
<td>Returns: The log level.</td>
<td></td>
</tr>
<tr>
<td><code>debugOn()</code></td>
<td>Determines if debug is turned on.</td>
<td>Parameters: N/A</td>
<td>Returns: (boolean) If true, debug is on; if false, debug is off.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Example</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Example</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>var gl = new GSLog(&quot;com.snc.sla.tasksla.log&quot;, &quot;TaskSLA&quot;); gl.setLevel(&quot;debug&quot;); gs.print(gl.getLevel());</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Output:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*** Script: debug</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>var gl = new GSLog(&quot;com.snc.sla.tasksla.log&quot;, &quot;TaskSLA&quot;); gl.setLevel(&quot;debug&quot;); gs.print(gl.debugOn());</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Output:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*** Script : true</td>
</tr>
</tbody>
</table>
//subtract 1 day
gdt.addDaysLocalTime(-1);

//add 3 weeks

gdt.addWeeksLocalTime(3);

//subtract 6 months.
gdt.addMonthsLocalTime(-6);

//add 1 year, representing the date and time using the UTC timezone instead of the local user's timezone.
gdt.addYearsUTC(1);

Reprocess received emails

It is possible to reprocess emails that were received by your instance.

A similar “Reprocess received emails” UI action gives a list choice option at the bottom of the email list, which can be used to reprocess multiple emails at once.

A UI action named “Reprocess Email” puts a button on the email form:

![Figure 951: Reprocess Email UI action](image)

The script and condition are duplicated below for easy cutting and pasting. This UI action gives you a button when you open an email (in System Logs --> Email) that has a type of "received" or "received-ignored". It will create an event that will reprocess that incoming email through the Inbound Email Actions.

**Condition**

```javascript
current.type == 'received' || current.type == 'received-ignored'
```

**Script**

```javascript
var evt = new GlideRecord('sysevent');
evt.initialize();
evt.process_on = gs.nowDateTime();
```
Sample ASP.NET with C Sharp redirect with cookies

This sample ASP.NET code creates a simple authentication portal and passes an unencrypted HTTP header as a cookie.

**Note:** Functionality described here requires the **Admin** role.

**Note:** Cookies are domain specific and cannot be used across different network domains. The only domain that can read a cookie is the domain that sets it. It does not matter what domain name you set. If you do not have an option of your SSO portal being in the same network domain as your ServiceNow instance (for example, in an on-premises deployment, an alternative is to pass the SSO token using URL GET or POST parameters.

This sample assumes:

- The web server supports ASP.NET and C#
- The target ServiceNow instance is `https://<instance name>.service-now.com/`
- SiteMinder or another single sign-on application has pre-authenticated the user
- The target ServiceNow instance expects an HTTP header of SM_USER

Change the ASP code to redirect users to the proper ServiceNow instance.

```html
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Portal Page Login</title>
    <%--    <meta http-equiv="REFRESH" content="0;url=https://<instance name>.service-now.com/">--%>
</head>
<body>
<form id="form1" runat="server">
<h2><b>Portal Page Login</b></h2>
<hr style="position: static" />
<br />
<asp:Label ID="Label2" runat="server" Font-Size="Larger" Height="21px" Style="position: static" Text="Instance URL:" Width="113px"></asp:Label>
<asp:TextBox ID="urlBox" runat="server" Font-Size="Large" Style="position: static"></asp:TextBox><br />
<br />
<asp:Label ID="Label1" runat="server" Font-Size="Larger" Height="17px" Style="position: static," Text="User Id:" Width="113px"></asp:Label>
<asp:TextBox ID="userNameBox" runat="server" Font-Size="Large" Style="position: static;">"</asp:TextBox><br />
<br />
<asp:Button ID="Button1" runat="server" Height="39px" Style="position: static;" Text="Ok" Width="88px" OnClick="Button1_click" />
</form>
</body>
</html>
```
Figure 952: ASP Portal Redirect

The following C# code handles the OnClick button event for the form. The code:

- Creates the cookie "SM_USER"
- Performs a redirect to the URL specified on the ASP form.

Change the C# code to create the proper cookie name.

```csharp
using System;
using System.Data;
using System.Configuration;
using System.Web;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_click(object sender, EventArgs e)
    {
        try
        {
            HttpCookie myCookie = new HttpCookie("SM_USER");
            myCookie.Value = userNameBox.Text;
            Response.Cookies.Add(myCookie);
            Response.Redirect(urlBox.Text);
        }
        catch {}
    }
}
```
Useful approval assignment scripts

This is a searchable version of the useful approval and assignment scripts.

Caution: The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

For an easy-to-navigate version, visit the Useful Scripts portal.

Assign a group for ESS requests
Type: Assignment Rule
Description: This script automatically assigns a group for all ESS Requests.
Script example:

```java
if(current.opened_by.roles==""){
    current.assignment_group.setDisplayValue('Network');
    current.update();
}
```

Assign Catalog Item to Group Based on Delivery Plan Task
Type: Assignment Rule
Description: This assignment rule assigns a service catalog item to the database group if it uses a delivery plan that has a catalog task assigned to the desktop group.

```java
//Return catalog items that have no group but do have a delivery plan assigned
var ri = new GlideRecord("sc_cat_item");
ri.addQuery("group","=",null);
ri.addQuery("delivery_plan","!=",null);
ri.query();
while(ri.next()){    
    gs.log("Found an item");
    //Return tasks that point to the same delivery plan as the above item
    var dptask = new GlideRecord("sc_cat_item_delivery_task");
    dptask.addQuery("delivery_plan","=",ri.delivery_plan);
    dptask.query();
    while(dptask.next()){    
        gs.log("Found a task");
        var gp = dptask.group.getDisplayValue();
        gs.log(gp);
        //If the task is assigned to desktop, assign the item's group to desktop
        if(dptask.group.getDisplayValue()=="Desktop"){
            ri.group.setDisplayValue("Desktop");
            gs.log("updating "+ ri.getDisplayValue());
            ri.update();
            break;
        }
    }
}
```

Assign items with one task
Type: Assignment Rule
Description: Automatically assigns any catalog items with only one task associated to a particular group.

```java
//Get the catalog item for the current requested item
var scCatItem = new GlideRecord("sc_cat_item");
if(scCatItem.get('sys_id', current.cat_item)){
```
Assignment based on workload
Type: Business Rule

Description: Populate the assigned to based on the assignment group member who has the least amount of active incidents.

Parameters:
- order: >1000 if you want to execute after assignment rules
- condition: current.assigned_to == " & current.assignment_group != "
- when: before, insert/update

```javascript
var assignTo = getLowestUser();
gs.addInfoMessage("assigning to is " + assignTo);
current.assigned_to= assignTo;

function getLowestUser(){
    var userList = new Array();
    var cg = new GlideRecord('sys_user_grmember');
    cg.addQuery('group', current.assignment_group);
    cg.query();
    while(cg.next()){var tech = cg.user.toString();
        var cnt = countTickets(tech);
        gs.addInfoMessage("Tech counts " + cg.user.name+ ' '+ cnt +" " + tech);
        userList.push({ sys_id: tech, name: cg.user.name, count: cnt });}
    for(var i=0; i < userList.length; i++){
        gs.addInfoMessage(userList[i].sys_id+" + userList[i].name+" + userList[i].count);
    }
    userList.sort(function(a, b){
        gs.addInfoMessage("Sorting: " + a.sys_id"(" + a.count");
        "+ b.sys_id"(" + b.count")");
        return a.count- b.count;});
    if(userList.length<=0)return"; return userList[0].sys_id;}

function countTickets(tech){
    var ct = new GlideRecord('incident');
    ct.addQuery('assigned_to',tech);
    ct.addQuery('active',true);
    ct.query();
    return ct.getRowCount();}
```

Run assignment rules when category is changed
Type: Client script
Table: Incident
Description: This example is an onChange client script on the category field within Incident. Note: this script used to use synchronous AJAX (asynchronous behavior is specified by the third parameter of the ajaxRequest call). The implementation below uses asynchronous AJAX. The drawback of using the synchronous version is that a network response problem could cause the browser to hang.

```javascript
// Make an AJAX request to the server to get who this incident would be
// assigned to given the current values in the record. This runs the
// rules that have been defined in System Policy and returns the assigned_to
// and
// the assignment_group

function onChange(control, oldValue, newValue, isLoading){
    if(isLoading){return;
    // No change, do not do anything
}

    // Construct the URL to ask the server for the assignment
    var url = "xmlhttp.do?sysparm_processor=AJAXAssignment&sys_target=incident";
    var uv = gel('sys_uniqueValue');
    if(uv){
        url +="&sys_uniqueValue="+ uv.value;
    // Make the AJAX request to the server and get the response
    var serial = g_form.serialize();
    // get all values currently assigned to the incident
    var response = ajaxRequest(url, serial, true, responseFunc);

    // This callback function handles the AJAX response.
    function responseFunc(response){
        var item = response.responseXML.getElementsByTagName("item") [0];
        // Process the item returned by the server
        if(item){
            // Get the assigned_to ID and its display value and put them on the form
            var name = item.getAttribute("name");
            var name_label = item.getAttribute("name_label");
            if(name_label && name){
                g_form.setValue('assigned_to', name, name_label);
            }else{
                g_form.setValue('assigned_to', '', '');
            }
            // Get the assignment_group ID and its display value and put then on the
            form
            var group = item.getAttribute("group");
            var group_label = item.getAttribute("group_label");
            if(group_label && group){
                g_form.setValue('assignment_group', group, group_label);
            }else{
                g_form.setValue('assignment_group', '', '');
            }
        }
    }
}
```

**Custom approval UI macro**

**Type:** UI macro

The following option illustrates how to obtain more detail from the **My Approvals** view of an Execution Plan by creating a new UI Macro.

- Navigate to **System UI** and click **UI Macros**.
- Rename the existing "approval_summarizer_sc_task" to something like "approval_summarizer_sc_task_old" and deactivate it.
- Create a new one using the same name ("approval_summarizer_sc_task"). The name should basically tell you what the macro does and to what it applies. In this case, we're replacing an existing one so we decided to re-use the existing name.
Then copy the xml script at the bottom of this article into the xml code window in the new UI Macro. This is a great way to give some detail to an approver when you are doing line item approvals using approval tasks within the Service Catalog Execution Plans.

### Different ways

#### Old way

This is the view you see in **My Approvals** when using an approval task using the old method.
Notice there is not much detail telling the approver what they are actually approving. You can see the short description of the task but not much information about the item.

**New way**

This is the view you will see if you use the xml script below in place of the OOB (out-of-box) UI Macro.

Using this method you can see details much like the request approval. You have a link into the item ordered, a short description (which contains the ability to expand the variables from the item), price, quantity and the total price. This helps the approver in that it shows more detail. They can now see what they are actually approving.

**Useful field scripts**

This is a searchable version of the useful field customization scripts.
**Caution:** The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

For an easy-to-navigate version, visit the Useful Scripts portal.

**AKA Incident Template, Auto Assignments, Quick Calls, Call Script, Auto Populate**

Let's say you want to auto-fill your **Short Description** based on the selected **Subcategory**. First, create a lookup table, then populate the key field, in this case the **Subcategory** and the auto-filled field, **Short Description**. So let's say your table had a record with **Subcategory** = Password and **Short Description** = Password Reset. When the user selects the **Subcategory** of Password on the Incident form, a client script looks up the matching record and sets **Short Description** equal to Password Reset.

**Client script settings...** Type = onChange, Table name = incident, Field name = Subcategory

```javascript
function onChange(control, oldValue, newValue, isLoading){
  if(isLoading){return;}
  var newrec = gel('sys_row');
  //Check if new record
  if (newrec.value==-1) {
    var lookup = new GlideRecord('u_short_desc_lookup');
    lookup.addQuery('u_subcategory', g_form.getValue('subcategory'));
    lookup.query();
    var temp; //temp var - reusable
    if(lookup.next()){
      temp = lookup.u_short_description;
      if(null!= temp) {
        //Set the form value from lookup if there is a lookup value
        g_form.setValue('short_description', temp);
      } else {
        g_form.setValue('short_description','""');
      }
    } else {
      //If a lookup record does not exist based on lookup.addQuery
      //Then set to UNDEFINED or NULL depending on type
      g_form.setValue('short_description','""');
    }
  }
}
```

You could populate multiple fields or even pull Call Script questions into the **Comments** field so call center personnel gather good information to pass on to a technician. There are already Assignment Rules, Templates, and Wizards built in that perform similar functions.

**Disable HTML tags in descriptions**

Description: This code disables HTML tags in descriptions and short descriptions by substituting the tags with harmless versions that won’t execute.

```javascript
function doit(){
  var desc = current.description.toString();
  var shdesc = current.short_description.toString();
  if(desc.indexOf('script>')>-1|| shdesc.indexOf('script>')>-1){
    desc = desc.replace(/<script>/g,"(script)"));
    current.description = desc.replace(/</script>/g,"(\</script")
    shdesc = shdesc.replace(/<script>/g,"(script)"));
    current.short_description = shdesc.replace(/</script>/g,"(\</script")
  }
}
```
Eliminate leading and trailing spaces in fields

Table: sys_user

Description: This example of the script trims trailing and leading spaces in the FirstName and LastName fields of sys_user.

doit();

function doit(){
    var gr =new GlideRecord('sys_user');
    gr.query();
    while(gr.next()){
        if((gr.first_name.toString().length!=
            gr.first_name.toString().trim().length)||(gr.last_name.toString().length!=
            gr.last_name.toString().trim().length)){
            gr.first_name= gr.first_name.toString().trim();
            gr.last_name= gr.last_name.toString().trim();
            gr.autoSysFields(false);
            gr.update();}
    }
}

Make a field label flash

Type: Client script

Description: The following example is for the number field on incident. The label will flash for two seconds:

    g_form.flash("incident.number","#FFFACD",0);

The arguments for the flash method are as follows:

1. tablename.fieldname
2. RGB color or acceptable CSS color like "blue" or "tomato"
3. integer that determines how long the label will flash:
   - Use 2 for a 1-second flash
   - Use 0 for a 2-second flash
   - Use -2 for a 3-second flash
   - Use -4 for a 4-second flash

Do not specify this argument if you want the field label simply colored the specified color.

Make field label bold

Type: Client script

Description: This script makes the label of a particular field (in this case, Short Description on the Incident Table) bold.

    functiononLoad(){
        var l = g_form.getLabel('incident.short_description');
        l.style.fontWeight='bold';
    }

Make fields read-only

Type: Client script

Table: Incident

Description: This onLoad client script makes fields read-only. For this example, the script makes the following fields on the Incident table read-only: Incident state, Impact, Urgency, Priority, Configuration
item, and Assigned to. It also removes the magnifying glass for the read-only Reference Fields (Configuration item and Assigned to).

```javascript
function onLoad(){
  var incidentState = g_form.getValue('incident_state');
  if( incidentState == '6' || incidentState == '7'){
    g_form.setReadonly('incident_state',true);
    g_form.setReadonly('impact',true);
    g_form.setReadonly('urgency',true);
    g_form.setReadonly('priority',true);
    g_form.setReadonly('cmdb_ci',true);
    g_form.setReadonly('assigned_to',true);}}
```

Set current date/time in field

Type: Client script

Description: You can use the following two lines to set the current date and time in a date/time field. This bypasses the problem of getting the value into the proper format and proper timezone.

```javascript
var ajax = new GlideAjax('MyDateTimeAjax');
ajax.addParam('sysparm_name','nowDateTime');
ajax.getXML(function(){
  g_form.setValue('put your field name here', ajax.getAnswer());});
```

System script include

```javascript
// Be sure the "Client callable" checkbox is checked

var MyDateTimeAjax = Class.create();
MyDateTimeAjax.prototype = Object.extendObject(AbstractAjaxProcessor, {
  nowDateTime: function()
  {
    return gs.nowDateTime();
  }});
```

Toggle timer field by field name

Type: Client script

Description: Toggles the timer field based on a particular field name.

```javascript
function toggleTimerByFieldName(fieldName){
  //Step 1: Find the timer object
  //timeObjectName: the timer objects name as it would normally be referenced
  //timeObjectHidden: the hidden input node in the field td
  //timeObjectParent: the parent td node containing the field and it's constituent nodes
  //timeObjectFields: anchor tag with onclick to stop timer

  var timeObjectName = fieldName;
  var timeObjectHidden = gel(timeObjectName);

  //Step 2: simulate click stop button
  var timeObjectParent;
  var timeObjectFields;

  //verify that we got the correct object
  if(timeObjectHidden.type=="hidden"){
    //Get Parent td node
    timeObjectParent = timeObjectHidden.parentNode;
```
//Get input fields
var timeObjectFields = timeObjectParent.getElementsByTagName("input");

//simulate click of stop button
var timerTestString = "paused";
var timerImg;

//loop through input objects looking for the pause timer object
for(var elIt=0; elIt < timeObjectFields.length; elIt++){
  if(timeObjectFields[elIt].id.match(timerTestString)){
    if(timeObjectFields[elIt].value=="false"){
      timeObjectFields[elIt].value="true";
      timerImg = timeObjectParent.getElementsByTagName("img")[0];
      timerImg.src="images/timer_start.gifx";
    }
    else if(timeObjectFields[elIt].value=="true"){
      timeObjectFields[elIt].value="false";
      timerImg = timeObjectParent.getElementsByTagName("img")[0];
      timerImg.src="images/timer_stop.gifx";
    }
  }
}

Modify GlideDateTime field value

Type: A server side script that accesses a GlideDateTime field.

Description: Given a GlideDateTime field or script object, show a variety of ways to easily modify value. The same concept also applies to the GlideDate object.

//You first need a GlideDateTime object
//this can be from instantiating a new object "var gdt = new GlideDateTime()"
//or getting the object from a GlideDateTime field
//getting the field value (for example: var gdt = current.start_date)
//only returns the string value, not the object
//to get the object use var gdt = current.start_date.getGlideObject();
//now gdt is a GlideDateTime object
var gdt = current.start_date.getGlideObject();

//All methods can use negative values to subtract intervals

//add 1 hour (60 mins * 60 secs)
gdt.addSecondsLocalTime(3600);

//add 1 day
gdt.addDaysLocalTime(1);

//subtract 1 day
gdt.addDaysLocalTime(-1);

//add 3 weeks
gdt.addWeeksLocalTime(3);

//subtract 6 months.
gdt.addMonthsLocalTime(-6);

//add 1 year, representing the date and time using the UTC timezone instead of the local user's timezone.
gdt.addYearsUTC(1);

Useful scheduling scripts

A business rule script specifies the actions that the business rule takes. Scripts commonly include predefined global variables to reference items in your system, such as the current record. Global variables are available to all business rules.
Caution: The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

Calculate duration given a schedule
Type: Before update/insert business rule
Table: Incident
Description: A Business Duration calculates the Open to Close duration on an incident based on the particular Schedules on page 1040. If there is no schedule specified, the script will simply use the first schedule returned by the query.

Script example:
The example below sets the resolved duration when the incident state moves to resolved.

```javascript
if(current.incident_state==6){
  var dur = calcDurationSchedule(current.opened_at,current.sys_updated_on);
  current.u_resolved_duration= dur;
}

function calcDurationSchedule(start, end){
  // Get the user
  var usr =new GlideRecord('sys_user');
  usr.get(gs.getUserID());
  // Create schedule - pass in the sys_id of your standard work day schedule and pass in the users timezone
  var sched =new GlideSchedule('08fcd0830a0a0b2600079f56b1adb9ae',usr.time_zone);
  // Get duration based on schedule/timezone
  return(sched.duration(start.getGlideObject(), end.getGlideObject()));}
```

Check upcoming termination dates
Type: Scheduled script
Description: This script checks nightly for termination dates on contracts coming up in 90, 50, or 10 days (depending on the contract duration field).

Script example:

```javascript
function contractNoticeDue(){
  var gr =new GlideRecord("contract");
  gr.addQuery("u_contract_status","Active");
  gr.query();
  while(gr.next()){ 
    if((gr.u_termination_date<= gs.daysAgo(-90))&&(gr.u_contract_duration=="Long")){
      gr.u_contract_status="In review";
    }else if((gr.u_termination_date<= gs.daysAgo(-50))&&(gr.u_contract_duration=="Medium")){
      gr.u_contract_status="In review";
    }else if((gr.u_termination_date<= gs.daysAgo(-10))&&(gr.u_contract_duration=="Short")){
      gr.u_contract_status="In review";
    }
    gr.update();
  }
}
```

Use scripts in business rules to accomplish common tasks such as:
- Comparing two date fields.
• Parsing XML payloads.
• Aborting a database action in a business rule.

With scripts, you can also:
• Specify the operation that triggers the business rule.
• Use the scratchpad with display business rules to change form values just before a user loads the form.
• Use the OR condition like you would in a condition builder.

You can also utilize the system's scripting functionality available for server-side scripts.

You can use options on the Business Rules form to build conditions, set field values, and display alert messages without needing to write a script.

Abort a database action in a business rule

During a before business rule script, you can cancel or abort the current database action using the current.setAbortAction(true) method.

For example, if the before business rule is executed during an insert action, and you have a condition in the script that calls current.setAbortAction(true), the new record stored in current is not created in the database.

Add autofill functionality

Add autofill functionality is also called incident template, auto assignments, quick calls, call script, or auto populate.

Let's say you want to auto-fill your Short Description based on the Subcategory selected. First, create a lookup table, then populate the key field, in this case Subcategory and the auto-filled field, Short Description. So let's say your table had a record with Subcategory = Password and Short Description = Password Reset. When the user selects the subcategory of Password on the Incident form a client script looks up the matching record and sets short description equal to Password Reset. Client script settings...

Type = onChange, Table name = incident, Field name = Subcategory.

```javascript
function onChange(control, oldValue, newValue, isLoading) {
    if (isLoading) { return; }
    var newrec = gel('sys_row');
    // Check if new record
    if (newrec.value == -1) {
        var lookup = new GlideRecord('u_short_desc_lookup');
        lookup.addQuery('u_subcategory', g_form.getValue('subcategory'));
        lookup.query();
        var temp; //temp var - reusable
        if (lookup.next()) {
            temp = lookup.u_short_description;
            if (null != temp) { // Set the form value from lookup if there is a lookup value
                g_form.setValue('short_description', temp);
            } else {
                g_form.setValue('short_description', "" );
            }
        } else {
            // If a lookup record does not exist based on lookup.addQuery
            // Then set to UNDEFINED or NULL depending on type
            g_form.setValue('short_description', "" );
        }
    }
}
```

You could populate many fields or even pull in call script questions into the Comments field so call center personnel gather good information to pass on to a technician. There are already Assignment Rule, Templates and Wizards built in that perform similar functions.
Example script: A default before-query business rule

You can use a query business rule that executes before the database query is made to prevent users from accessing certain records.

**Caution:** The customization described here was developed for use in specific instances, and is not supported by ServiceNow. This method is provided as-is and should be tested thoroughly before implementation. Post all questions and comments regarding this customization to our community forum.

Consider the following example from a default business rule that limits access to incident records.

**Table 1387: Default business rule limits access to incident records**

<table>
<thead>
<tr>
<th>Name</th>
<th>Table</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>incident query</td>
<td>Incident</td>
<td>before, query</td>
</tr>
</tbody>
</table>

**Example script**

This example prevents users from accessing incident records unless they have the itil role are listed in the Caller or Opened by field. So, for example, when self-service users open a list of incidents, they can only see the incidents they submitted.

```javascript
if(!gs.hasRole("itil")&& gs.isInteractive()){
    var u = gs.getUserID();
    var qc = current.addQuery("caller_id", u).addOrCondition("opened_by", u).addOrCondition("watch_list","CONTAINS", u);
    gs.print("query restricted to user: "+ u);}
```

**Note:** You can also use access controls to restrict the records that users can see.

**Schedule script for weekdays**

Type: Business Rules/Client Scripts

This script schedules the script for weekdays. Insert any script where it says "Your Script Here."

```javascript
var go = 'false';
var now = new Date();

// Correct time zone, which is by default GMT -7
now.setHours(now.getHours()+8);
var day = now.getDay();

// No go on Saturday or Sunday
if(day !=0&& day !=6){
    // (your script here)
}
```

**Set date field according to current date**

This script sets a date field depending on the current day of the week. In this example, if the day is Monday through Wednesday, it sets the date to this coming Monday; otherwise it sets the date field to next Monday.

```javascript
function setCabDate(){
```

© 2017 ServiceNow. All rights reserved.
var today = new Date();
var thisDay = today.getDay();
// returns 0 for Sunday, 1 for Monday, etc. thru 6 for Saturday.
var thisMon = new GlideDateTime();
thisMon.setDisplayValue(gs.beginningOfThisWeek());
var nextMon = thisMon.getNumericValue();
nextMon += (1000 * 60 * 60 * 24 * 7);

if((thisDay < 4) && (thisDay > 0))
    // if today is Mon thru Wed (thisDay = 1, 2, or 3), set cab to this coming Monday.
    current.u_req_cab_rev_date.setDateNumericValue(thisMon.getNumericValue());
elseif((thisDay >= 4) || (thisDay == 0))
    // if today is Thurs thru Sun (thisDay = 4, 5, 6, or 0), set cab to next Monday.
    current.u_req_cab_rev_date.setDateNumericValue(nextMon);

To validate the input of all date/time fields, you can use the following in a validation script (System Definition > Validation Scripts). Because the date/time format is hard coded in this script, it must match your instance’s date/time format. If your instance’s date/time format changes, you must update your validation script.

Set the validation script's type to Date/Time. Then, with this validation script, if a user enters an incorrect format in a date/time field, they will receive an error message.

function validate(value){
    // empty fields are still valid dates
    if(!value)return true;

    // We "should" have the global date format defined always defined. but there's always that edge case...
    if(typeof g_user_date_time_format !== 'undefined') return isDate(value, g_user_date_time_format);

    // if we don't have that defined, we can always try guessing
    return parseDate(value) !== null;
Sample ASP Script for unencrypted single sign-on

This sample ASP .NET code creates a simple authentication portal and passes an unencrypted HTTP header as a URL parameter.

This sample assumes:

- The web server supports ASP .NET
- The target instance is https://<instance name>.service-now.com/
- SiteMinder or another single sign-on application has pre-authenticated the user
- The target instance expects an HTTP header of SM_USER

Change the ASP code to redirect users to the proper instance and create the proper HTTP header.

```html
<html xmlns = "http://www.w3.org/1999/xhtml" >
<head runat = "server" >
<title >Portal Page Login </ title > <%--    <meta http-equiv = "REFRESH" content = "0;url=https://<instance name>.service-now.com/">--%>
<script runat = "server" >
    public void go_to(object sender, EventArgs e)
    {
        //Send URL parameters
        String URL = urlBox.Text + "?SM_USER=" + userNameBox.Text;
        Response.Redirect(URL);
    }
</script>
</head >
<body >
<form id = "form1" runat = "server" >
<h2 >Portal Page Login </ h2 >&lt; / h2 &gt;&lt; hr style = "position: static" / &gt;&lt; br / &gt;
```

Figure 953: Date/time validation
Validate date and time

To validate the input of all date/time fields, you can use the following in a validation script (System Definition > Validation Scripts).

Because the date/time format is hardcoded in this script, it must match your instance’s date/time format. If your instance’s date/time format changes, you must update your validation script.

Set the validation script’s type to "glide_date_time". Then, with this validation script, if a user enters an incorrect format in a date/time field, they will receive an error message.

```javascript
function validate (value ) { if ( !value ) { return true ; } return (getDateFromFormat (value , 'yyyy-MM-dd HH:mm:ss' ) != 0 ) ; }
```
Figure 955: Date/time validation

Workflow use case

Often you may need to provide users with a way to specify when a task or process is due. Using the DurationCalculator script include, you can calculate the due date, using either a simple duration or relative duration.

For information on schedules, which you can use as inputs to DurationCalculator methods, see Schedules on page 1040.

See the ServiceNow Developers site for API information.

This script demonstrates how to use DurationCalculator to compute a due date.

```javascript
/**
 * Demonstrate the use of DurationCalculator to compute a due date.
 *
 * You must have a start date and a duration. Then you can compute a
 * due date using the constraints of a schedule.
 */

gs.include('DurationCalculator');
executeSample();

/**
 * Function to house the sample script.
 */function executeSample(){

    // First we need a DurationCalculator object.var dc =new
    DurationCalculator();

    // --------------- No schedule examples ------------------
```
// Simple computation of a due date without using a schedule. Seconds are added to the start date continuously to get to a due date.
// Start in the middle of the night (2:00 am) and compute a due date 1 hour in the future. Without a schedule this yields 3:00 am.
dc.setStartDateTime("5/3/2012 02:00:00");if(!dc.calcDuration(3600)){
    gs.log("*** Error calculating duration");return;
}gs.log("Middle of night + 1 hour (no schedule): "+
dc.getEndDateTime());// "2012-05-03 03:00:00" three hours later

// Start in the middle of the night (2:00 am) and compute a due date 1 hour in the future. Since we start at 2:00 am the computation adds the 1 hour from the start of the day, 8:00am to get to 9:00am
// Setting the timezone causes the schedule to be interpreted in the specified timezone. Run the same code as above with different timezone.
dc.setTimeZone("GMT-2");
dc.setStartDateTime("5/3/2012 15:00:00");for(var i=2; i<24; i+=1){if(!dc.calcDuration(i*3600)){
    gs.log("*** Error calculating duration");return;
}gs.log("add " + i + " hours gives due date (GMT-2): "+
dc.getEndDateTime());}

// Setting the timezone causes the schedule to be interpreted in the specified timezone. Run the same code as above with different timezone. Note that the 8 to 5 workday is offset by the two hours as specified in our timezone.
dc.setTimeZone("US/Eastern");
dc.setStartDateTime("5/3/2012 15:00:00");for(var i=2; i<24; i+=1){if(!dc.calcDuration(i*3600)){
    gs.log("*** Error calculating duration");return;
}gs.log("add " + i + " hours gives due date (US/Eastern): "+
dc.getEndDateTime());}

/**
 * Add a specific schedule to the DurationCalculator object.
 * @param durationCalculator An instance of DurationCalculator
 */
function addSchedule(durationCalculator){//  Load the "8-5 weekdays excluding holidays" schedule into our duration calculator.
var scheduleName ="8-5 weekdays excluding holidays";
var grSched =new GlideRecord('cmn_schedule');
}}
Calculating a relative duration

An example of a relative duration calculation script.

This script calculates the relative duration for "Next day at 4pm if after 10am":

```javascript
// Next day at 4pm if before 10am
var days = 1;
if (calculator.isAfter(calculator.startDateTime, "10:00:00"))
    days++;

calculator.calcRelativeDueDate(calculator.startDateTime, days, "16:00:00");
```

This script demonstrates how to use DurationCalculator to calculate a relative duration.

```javascript
/**
 * Sample use of relative duration calculation.
 *
 */
gs.include('DurationCalculator');
executeSample();

/**
 * Function to house the sample script.
 */
function executeSample(){
    // First we need a DurationCalculator object. We will also use//
    the out-of-box relative duration "2 bus days by 4pm"var dc = new
    DurationCalculator();var relDur = "3bf802c20a0a0b52008e2859cd8abcf2";// 2
    bus days by 4pm if before 10am
    addSchedule(dc);

    // Since our start date is before 10:00am our result is two days from//
    now at 4:00pm.
    dc.setStartDateTime("5/1/2012 09:00:00");
    if (!
        dc.calcRelativeDuration(relDur)){
        gs.log("*** calcRelativeDuration failed");
        return;
    }
    gs.log("Two days later 4:00pm: " + dc.getEndDate());

    // Since our start date is after 10:00am our result is three days from//
    now at 4:00pm.
    dc.setStartDateTime("5/1/2012 11:00:00");
    if (!
        dc.calcRelativeDuration(relDur)){
        gs.log("*** calcRelativeDuration failed");
        return;
    }
    gs.log("Three days later 4:00pm: " + dc.getEndDate());

    // Add a specific schedule to the DurationCalculator object.
    //
    * @param durationCalculator An instance of DurationCalculator
    */
    function addSchedule(durationCalculator){
        // Load the "8-5 weekdays excluding holidays" schedule into our duration calculator.
        var scheduleName = "8-5 weekdays excluding holidays";
        var grSched = new GlideRecord('cmn_schedule');
        grSched.addQuery('name', scheduleName);
        grSched.query();
        if (!grSched.next()){
            gs.log('*** Could not find schedule "' + scheduleName + '"");
            return;
        } else {
            durationCalculator.setSchedule(grSched.getUniqueValue(), "GMT");
        }
    }
```
Calculating a simple duration

This business rule and script example demonstrate how to calculate a simple duration.

```javascript
var dur = new DurationCalculator();
dur.setSchedule(current.schedule);
dur.setStartDateTime('');

if(current.duration_type==''){
  dur.calcDuration(current.duration.getGlideObject().getNumericValue()/1000);}
else{
  dur.calcRelativeDuration(current.duration_type);
}

current.end_date_time= dur.getEndDateTime();
current.work_seconds= dur.getSeconds();
```

This script demonstrates how to use DurationCalculator to calculate a simple duration.

```javascript
/**
 * Sample script demonstrating use of DurationCalculator to compute simple durations
 *
 */
gs.include('DurationCalculator');
executeSample();

/**
 * Function to house the sample script.
 */
function executeSample(){

  // First we need a DurationCalculator object.
  var dc = new DurationCalculator();

  // Compute a simple duration without any schedule. The arguments can also be of type GlideDateTime, such as fields from a GlideRecord.
  var dur = dc.calcScheduleDuration("5/1/2012","5/2/2012");
  gs.log("calcScheduleDuration no schedule: "+ dur); // 86400 seconds (24 hours)

  // The above sample is useful in limited cases. We almost always want to use some schedule in a duration computation, let's load a schedule.
  addSchedule(dc);

  // Compute a duration using the schedule. The schedule specifies a nine hour work day. The output of this is 32400 seconds, or a nine hour span.
  dur = dc.calcScheduleDuration("5/23/2012 12:00","5/24/2012 12:00");
  gs.log("calcScheduleDuration with schedule: "+ dur); // 32400 seconds (9 hours)

  // Compute a duration that spans a weekend and holiday. Even though this spans three days, it only spans 9 work hours based on the schedule.
  dur = dc.calcScheduleDuration("5/25/2012 12:00","5/29/2012 12:00");
  gs.log("calcScheduleDuration with schedule spanning holiday: "+ dur); // 32400 seconds (9 hours)

  // Use the current date time in a calculation. The output of this is dependent on when you run it.
  var now = new Date();
  dur = dc.calcScheduleDuration("5/15/2012", new GlideDateTime());
```
gs.log("calcScheduleDuration with schedule to now: "+ dur);// Different on every run.}

/**
 * Add a specific schedule to the DurationCalculator object.
 * @param durationCalculator An instance of DurationCalculator
 */

function addSchedule(durationCalculator){// Load the "8-5 weekdays excluding holidays" schedule into our duration calculator.

var scheduleName = "8-5 weekdays excluding holidays";
var grSched = new GlideRecord('cmn_schedule');
grSched.addQuery('name', scheduleName);
grSched.query();

if(!grSched.next()){
    gs.log('*** Could not find schedule "+ scheduleName +"');
    return;
}

durationCalculator.setSchedule(grSched.getUniqueValue());

Elapsed time vs work time

Typically, setting a due date requires that you calculate the actual work time required for completion rather than the total time that elapses until the due date.

SLAs, Workflow tasks, and approvals are examples of situations that require the actual work rather than the total time that has elapsed. In this case, only the part of the day when work is performed is considered when determining when the work is to be complete. For example, if a task is due in 10 hours, but the actual time to perform the task is restricted to a business day schedule, the 10 hours of work can only be done at most 8 hours each day. If this work starts at 10am on Monday, it is expected to complete, or be due, on Tuesday at 12pm:

10am-5pm on Monday (6 hours) + 8am-12pm on Tuesday (4 hours)

How to implement a relative duration

The general steps for implementing a relative duration.

1. Create the cmn_relative_duration table.
2. Create the DurationCalculator script include.
3. Create a sample relative duration entry (for example, "Next business day by 4pm").
4. Add the needed fields to SLA tables to support relative durations.
5. Modify duration calculation for SLAs.
6. Modify SLA Percentage timer calculation for SLAs (this must use work_seconds).
7. Add schedule fields to the Workflow: Schedule and Timezone (selected based on the field from workflow table).
8. Add duration support fields to the Workflow Task activity.
9. Implement duration calculation script for the task activity.

Simple duration vs relative duration

How much work is required to complete a task can be expressed as a "relative duration".

Relative duration determines the expected due date and time relative to the starting time. Examples of relative durations include "Next business day by 4pm", or "2 business days by 10:30am".

To calculate a relative duration, the calendar and time zone must be considered to determine what "next business day" means since it is the calendar that defines which days are valid work days and the time zone will affect the result as well. As an example, consider "Next business day by 4pm":

- If it is Monday at 12pm: Next business day by 4pm => Tuesday at 4pm
• If it is Friday at 2pm: Next business day by 4pm => the following Monday at 4pm

**Note:** Next business day is often defined by a starting day and time. For example, "next business day at 4pm if before 2pm" indicates that if the current time is after 2pm on a business day, then "Next business day" really means 2 business days since today does not count.

### The relative duration table and the DurationCalculator methods

The cmn_relative_duration table supports the definition of a due date as either a duration of time or a relative duration.

This table consists of two fields: "name" and "script". The "script" field contains the relative duration calculation script. This script includes the "calculator" variable, which is used to calculate the due date.

The DurationCalculator script include can be used to perform the duration calculations. The following are methods that are available in this script include.

See the [ServiceNow Developers site](https://developer.servicenow.com) for API information.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setSchedule(String schedID, [String timezone])</td>
<td>Sets the schedule and time zone to be used for calculating the due date.</td>
</tr>
<tr>
<td>setStartDateTime(GlideDateTime start)</td>
<td>Sets the start time for the duration calculations. If 'start' is blank, uses current date/time.</td>
</tr>
<tr>
<td>calcDuration(int seconds)</td>
<td>Calculates the end date and time. Upon completion the this.endDateTime and this.seconds properties will be set to indicate the results of the calculation.</td>
</tr>
<tr>
<td>calcRelativeDuration(String relativeDurationID)</td>
<td>Calculates the duration using the specified relative duration script. Upon completion the this.endDateTime and this.seconds properties will be set to indicate the results of the calculation.</td>
</tr>
<tr>
<td>getEndDateTime()</td>
<td>Gets the this.endDateTime property that was set by calcDuration/calcRelativeDuration indicating the end date and time for the duration.</td>
</tr>
<tr>
<td>getSeconds()</td>
<td>Gets the this.seconds property that was set by calcDuration/calcRelativeDuration indicating the total number of seconds of work to be performed for the duration.</td>
</tr>
<tr>
<td>getTotalSeconds()</td>
<td>Gets the this.totalSeconds property that was set by calcDuration/calcRelativeDuration indicating the total number of seconds between the start and end times of the duration.</td>
</tr>
</tbody>
</table>

**Note:** This is the total work time, not the total time between start and end times and may be used to determine percentages of the work time.
The following functions are used in relative duration scripts:

Table 1389: Relative duration script functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isAfter(GlideDateTime dt, String time)</td>
<td>Is 'time' of day after the time of day specified by 'dt'? dt, if blank, uses current date/time. time is in &quot;hh:mm:ss&quot; in 24-hour format.</td>
</tr>
<tr>
<td>calcRelativeDueDate(GlideDateTime start, int days, String endTime)</td>
<td>Calculates the due date starting at 'start' and adding 'days' using the schedule and time zone. When we find the day that the work is due on, set the time to 'endTime' of that day. Upon completion, this.endDateTime and this.seconds properties will be set to indicate the results of the calculation. If endTime is blank, use end of the ending work day.</td>
</tr>
</tbody>
</table>
Index

Special Characters

.net 3375

A

access
  workflow scratchpad
  from business rules 3985
access account
  LDAPS 1881
access control 2498, 2530
access controls 2498, 2530
access from CMS pages 2356
account
  Twilio
  associating 2969
ACL 2498
ACLs
  client-callable script includes 2506
  configuration watcher
    execution plan 2512
    using 2513
  controlling whether script conditions apply to reference fields 2523, 2523
  creating ACL rules 2515
  debugging 2525
  evaluate the admin override at the access level 2507
  field ACL rules 2504
  field level debugging 2525
  grant or deny access 2519
  hasRoles() 2527, 2529, 2529
  in scoped applications 2506
  information sent to UA 2530
  internal and external users 2527
  mandatory roles 2527, 2529, 2529
  match ACL rules to objects 2520
  multiple ACL rules 2523
  permission requirement evaluation 2521
  process order for record ACL rules 2502
  processor rules 2504
  provide access to users 2529
  record ACL rules 2504
  snc_internal 2527, 2529, 2529
  table ACL rules 2505
  troubleshooting 2526
  UI page ACL rules 2505
action type
  inbound email, matching 2827
actions
  menu 2206
activate plugins 1233, 1233
activated by maint 1233
activating the Explicit Roles plugin 2529
activating the restore deleted records plugin 1622
active directory application mode (ADAM)
  access account 1881
  backup 1879
configure 1875
console 1877
container 1877
create objects 1877
delegation 1877
dependencies 1874
group objects 1877
instance 1875
LDAPS 1879
LDAPS certificate 1880
LDAPS connection 1881
objects 1877
organizational unit 1877
permission 1877
public key certificate 1880
recovery 1879
redundancy 1879
security 1874
testing 1878
troubleshooting 1878
user objects 1877
userproxy objects 1877
active Directory Federation Services
  creating a SAML logout 1938
  identity provider 1930, 1930, 1933, 1933, 1933, 1933, 1938, 1938, 1939
Active Directory, configure SSL access 1886
active transactions
  killing 2005
  viewing 2005
activities list
  customize 767
activity formatter
  activity filter configuration 107
  activity filter in UI1 108
  activity filter in UI16 and UI15 107
  add to a form 766
  create 769
  email property 769
  live feed 766
  security 764
ADAM
  ADAMSync 1881, 1882, 1882, 1883, 1884
  configuration file 1883, 1884
  configuring 1875
dependencies 1874
dependencies 1874
example 1884
instance 1875
security 1874
set up 1882
user accounts 1882
add
  actions
    chat window menu 2307
  field
    table 783
add blank line
  email template 2618
add item 1787
add role
   existing role 1723
add role to every user 3989
adding
elements 3399
   scrolling 3399
field
   service catalog checkout 3986
HTML fields
   media 806
media
   HTML fields 806
new department 1749
news
   scroller 3401
news panel
   scrolling 3400, 3401
scroller
   news 3401
scrolling
elements 3399
   news panel 3400, 3401
suggestion fields 952
ADFS 1930, 1930, 1933, 1933, 1933, 1938, 1938, 1939
   See also Active Directory Federation Services
ADFS integration 1930
administrator
table administration
   unique record identifier 1632
unique record identifier 1632
Administrator 295
administer ServiceNow 295
administering
context
   workflow 3643
   Legacy Notify 2968
table UI 1231
workflow
   context 3643
administration 295
administrative roles 1722
advanced reference qualifier
   use 1756
advanced search
   perform 2400
Aggregate API
   examples 3126
   GET 3119
AJAX 3926
AJAXClientTiming 3931
AJAXEvaluate processor
   enable 2459
all survey results
   view 527
altiris integration 1687
Amazon EC2
   instance 3789, 3789, 3792
annotate
   forms 716
annotate forms 716
APIs
cost management 3957
   Glide Server 3957
Apple home screen
custom icon, add 1178
application menu
   creating 47
      modules, creating 47
application menus
   adding to a category 46
application menus or modules
   disabling 44
      enabling 44
application navigator
   navigation filter 44
      UI11 40, 41
UI11 perspectives 42
UI15 36, 39
UI15 favorites 40
UI15 perspectives 40
UI16 31, 33
UI16 favorites
   editing 36
UI16 navigation history 37, 38
applications
   create a custom table 1616, 1622
apply ACLs
   AJAXGlideRecord 2459
Approval Action
   Input Variables 3680
   Results 3680
   States 3680
Approval and Rollback Activities 3680
Approval Coordinator
   Conditions 3682
   Input Variables 3682
   Result Values 3682
   States 3682
approval engines
   set up 452
Approval Group
   Conditions 3685
   Input Variables 3685
   Result Values 3685
   States 3685
approval notifications
   receiving 469
approval rules
   gating approvals 455
      process guide 459
Approval Rules
   approval rule 455
      gating approvals 455, 457
      set up 455, 457
approval summarizer
   changing 463
      formatter example 461, 762
      new custom, create 464
approval ui macro 3992
Approval User
   Conditions 3691
   Input Variables 3691

© 2017 ServiceNow. All rights reserved. 4029
Results 3691
States 3691
approvals
delegate 3800
generate
using approval rules 468
using approvers related list 468
using workflows 468
status 467
approve execution plan
use script 3898
assessable record
delete 350, 449
delete stakeholder 357
assessable records
generate 338
manage
automatically 328
managing
manually 329
Assessable Records 348, 449
assessment
take 446
assessment groups 323
assessment metric
assessable record
manage 328
category
delete 329
assessment metrics
assessment method 315
creating 330
data types 317
metric definitions, creating 337
metric templates 351
metric templates, creating 350
metric values, updating 351
script method 315
Assessment notification
workflow 371
assessment properties
editing 568
assessment questionnaires 448
assessment scorecard 384
assessment trigger condition
creating 363
example 367
assessments
appearance, customize 567
categories, create 325
category users 352, 1733
category users, creating 352
change the order of an answer 338
clean up data 361
create metrics 338
create quizzes with forms 411, 624
creating quizzes with forms 397, 411, 412, 420
email notifications 368
export 347
generate 323
generate assessable records 338
generate on-demand assessments 358
generate with on-demand API 359
import 347
installed components
business rules 304
client scripts 300
properties 298
script includes 300
tables 296
user roles 299
instances 362
metric categories 324
on-demand assessments 358
on-demand for multiple assessable records 359
on-demand for one assessable record 358
process 312
publish metric types 360
results 376, 379
roles 312
schedule types 360
scheduled
configure for 360
generate manually 361
vendor type, generating 361
scheduled assessments 360, 361
signatures 347
stakeholders 352, 1733
stakeholders, creating 353
view instances 362
weight categories and metrics 324
assessments overview
gauges 377
roles 377
assessments overview module 377
associating
account
Twilio 2969
Twilio account 2969
async business rules 2539
attachment
configuring
icon 751
disabling
table 749
file size
limiting 750
icon
configuring 751
table
disabling 749
viewing 749
Attachment API
DELETE 3162
GET 3146
GET file 3153
GET meta 3150
Large attachments 3163
POST 3154
POST multipart 3158
POST multipart example 3161
role requirements 3215
## Attachment Indexing
- enabling 752, 753, 1104
- attachment logging 751

## Attachments
- administering 749
- events 751
- restrict file extensions 750
- hide attachment [view] link 750

## Audit Roles
- remove 128
- rename 128
- uploading 127, 129

## Beam
- administering 2200
- icons
- creating 2206
- modifying 2206
- indicators
- creating 2201
- modifying 2201
- menu actions
- creating 2208
- modifying 2208
- related items
- creating 2204
- modifying 2204

## Bubble Chart
- components 381, 384

## Bubble Charts
- create 380
- view 379

## Build Tests Verify Functionality
- Scripted REST APIs 3232

## Business Rules
- creating 3853
- how they work 3850
- installed with live feed 2349
- scoped applications 3852

## Business Service Management Map
- annotation 2226
- collapsed objects, viewing 2227
- component nodes 2230
- default map indicators 2228
- past outages, managing 2229

## Business Service Management Map
- installed with 2196
- next generation version 2180
- use a bsm map 2212, 2213, 2217, 2219, 2220, 2220, 2221, 2222, 2223, 2223, 2224, 2225, 2225, 2226

## Business Service Management Map
- Legacy 2194

## C
- c sharp integration 3375
- Calculated Currency 979
- calculated price fields 957

## Calendars
- creating 29
- customizing 29
- deleting 30
- opening 30

## Brand Instance
- 257, 261

## breadcrumbs
- restricting 691
- suppressing 691

## breadcrumbs
- restricting 691
- suppressing 691

## Breadcrumb
- restricting 689

## Base System Roles
- special administrative roles 1728

## Blackberry
- faq 1200

## Bookmarks
- creating 29
- customizing 29
- deleting 30
- opening 30

## Bookmarks
- creating 29
- customizing 29
- deleting 30
- opening 30

## backup
- active directory application mode (ADAM) 1879
- ADAM 1879

## Banner Frame
- UI11 26
- UI15 21
- UI16 18

## Base System Roles
- special administrative roles 1728

## Basic Configuration
- 257, 261

## Blackberry
- faq 1200

## Bookmarks
- creating 29
- customizing 29
- deleting 30
- opening 30

## Brand Instance
- 257, 261

## Breadcrumbs
- restricting 691
- suppressing 691

## Breadcrumb
- restricting 689

## BSM map
- administering 2200
- icons
- creating 2206
- modifying 2206
- indicators
- creating 2201
- modifying 2201
- menu actions
- creating 2208
- modifying 2208
- related items
- creating 2204
- modifying 2204

## Bubble Chart
- components 381, 384

## Bubble Charts
- create 380
- view 379

## Build Tests Verify Functionality
- Scripted REST APIs 3232

## Business Rules
- creating 3853
- how they work 3850
- installed with live feed 2349
- scoped applications 3852

## Business Service Management Map
- annotation 2226
- collapsed objects, viewing 2227
- component nodes 2230
- default map indicators 2228
- past outages, managing 2229

## Business Service Management Map
- installed with 2196
- next generation version 2180
- use a bsm map 2212, 2213, 2217, 2219, 2220, 2220, 2221, 2222, 2223, 2223, 2224, 2225, 2225, 2226

## Business Service Management Map
- Legacy 2194

## C
- c sharp integration 3375
- Calculated Currency 979
- calculated price fields 957

## Calendars
- creating 29
- customizing 29
- deleting 30
- opening 30

## Call
- conference
- viewing 2966
canceling workflow 3646
Catalog homepage managing 502, 502, 502, 504, 506, 507
categories creating in survey management 548
survey management 549
categories for assessments 325
Category 3365, 3367
category results 435
category users 352, 1733
category users, creating 352
certificate certificate authority (CA) 2452
criteria 2451
Distinguished Encoding Rules 2451
generate with keytool 2452
Privacy Enhanced Mail 2451
trust 2452
thanked server 2453
upload 2450
validate 2454
certificates ldap 2449
ldap client 2452
mutual authentication 2452
change management business rule 1708
change management integration
Outlook 1707
Outlook, turning on 1709
change password module
create 201
chat activate 2301
Chat
installed components
fields 2302
script includes 2303
chat action details 2309
chat actions 2307
chat room access rights
define 2305
chat room message read access
change 2305
chat window menu
adding actions 2307
checklists enable 130
using with visual task boards
configuring checklists 2250
creating checklists from templates 2249
creating new checklists 2248
deleting checklists 2249
child record 728
choice Legacy Notify
creating 2964
response 2964
choice list
time zone 1055
CI relationships and history 2567
class
base 1417
child 1417
parent 1417
ClassName
methodName() 2945, 2947, 2950
Client
call(Object identifier) 2954
Client(Object) 2953
forwardCall(Object argument) 2956
hangupCall() 2954
mute(Boolean muted) 2956
Notify WebRTC Client 2953
pickupCall() 2955
client scripting 3900
client scripts
service catalog 3907
Service Catalog 3906, 3908, 3908
UI scripts 3913, 3913, 3914
variable fields, access to 3902
Client Scripts
Service Catalog 3907
Client Transaction Timings 1002, 1003, 1004, 2554
CMDB Import
configuration 1701
import set data 1701
web services 1700
CMS gauge
on a page, putting 2070
what happens on a click, controlling 2070
CMS gauge support 2070
CMS translation
viewing 2122
coalesce field
using 1542
coalescence 1544
Common UI Elements
reference icon 150
Company Profile
create 271, 273, 277, 277, 278, 279, 280
complexity limits
joins 1416
computer telephony integration 1711
condition
parameters 2210
condition activities
if 3703
switch 3704
wait for condition 3708
wait for WF event 3709
condition builder
date/time fields 158
empty fields, filtering 158
format 157
condition builders
operators 159
condition count 859
condition count widget
add 860

© 2017 ServiceNow. All rights reserved. 4032
conference call viewing 2966
configure group types assignment groups 1755
configure mobile location and barcode scanning 1228
configure transaction quota rule 2035
configuring attachment icon 751
endpoints Twilio 2971
icon attachment 751
Legacy notify Twilio 2969
suggested text
journal fields 954
string fields 953
Twilio endpoints 2971
Legacy notify 2969
view rules 1232
configuring email alternate
POP3 2707
SMTP 2705
SMTP and POP3 2709
standard, POP3 2705
standard, SMTP 2705
configuring lists 672, 673, 674, 676, 678, 679, 679
connect administering 2282
chat mini windows
drag and drop files 2281
connect actions 2283
connect interface 2267
connect overlay disabiling 2283
connect sidebar 2269
connect workspace conversation pane 2275
collection tools 2275
installed components
business rules 2264
notifications 2264
properties 2263
tables 2262
mentioning users 2281
mini windows 2271
notifications 2277, 2278
polling interval 2282
record conversations
disabling for all tables 2290
disabling for specific tables 2290
supported browsers 2262
upload a profile picture 2280
visual task boards
sharing 2258, 2258, 2281, 2288
connect chat administering 2288
connect workspace
adding and removing conversation members 2288
editing conversation details 2287
conversation members
adding and removing 2288
email notification interval 2289
following a record 2286
record conversations enabling 2289
starting conversations 2286
using 2286
Connect chat
browser notifications 2278
Connect Chat
activate 2262
Connect chat notifications
browser support 2278
connect support
activating 2265
administering 2296
administering queues 2297
creating an incident from a conversation 2296
escalating a conversation 2295
installed components
business rules 2267
properties 2266
tables 2265
making a queue accessible to end users 2299
monitoring incoming conversations 2291
sharing knowledge 2292
transferring a conversation 2293
Connect Support
chat timeout 2299
Connect Support migration 2297
connection
LDAPS 1881
testing 1881
connection test module
add 2029
container
ADAM 1877
create 1877
content blocks
create header block 2074
dynamic block 2092
static html block 2095
Content Blocks
add an image 2096
catalog cart block 2093
content management 2099
Flash movie block 2096
Live Feed 2099
static html block 2096
content management
content types 2068
corporate style guide 2044
design 2041
organize content 2042
prepare data 2044
prototypes 2046
rapid web design 2046
team members 2043
Content Management
administration 2120
and the Apache Jelly engine 2100
business rules 2050
client script 2050
configure list definitions 2095
configure menu items 2090
configuring the content type 2095
content blocks
dynamic blocks 2092
content links 2097
content site 2051
content types 2069
Content Types 2071
copy a page to create a template 2066
copy a site 2056
create a navigation menu block 2076
create menu sections 2087
creating menu items 2091
creating menu sections 2089
customizing the menu style 2082
DIV-based layouts 2110
global search 2121
installed with 2048
integration points 2111
integration points example 2118
Jelly Examples 2103
list block 2093
master template 2065
menu types 2064
navigation 2046
navigation menu example 2081
roles 2049
script includes 2049
security 2056, 2057, 2064, 2108
source view 2114
style
themes 2107
tables 2048
testing 2120
translation 2121
use content blocks 2072, 2072, 2073, 2074, 2093, 2095, 2096, 2098
Content Management Integration Points
dynamic methods 2117
manage system content 2046
content management meta tags
hierarchy 2110
page level 2109
site level 2110
content management system 2040
Content Management System
activate 2048
Content Management templates 2065
content pages
add content 2061
add to update sets and applications 2064
assign to a site 2064
copy 2064
create 2058
content types
creating 2069
Content Types
view 2071
content-sensitive help
help context types 191
context
workflow
administering 3643
context menus
action script 294
create 291
dynamic actions script 294
list header 288
list row 289
list title 287
onshow script 295
context ranking
definition, creating 697
list, ranking 698
plugin, activate 696
sort order, apply 700
update sets 697
context-sensitive help
accessing from the UI 186
administering 187
creating custom contexts 187
help context 192
introduction to 186
prioritization 192
setting base URL for custom help 190
system properties for 189
contextual search
add functions 3009
associate with wizard 3021
define search context 3010
defining 3009, 3009, 3019, 3021
installed components 3036, 3038, 3039, 3041, 3043
record producer 3019
Contextual Search
administering 3034
Administrating 3032
configure properties 3032
Defining 3012, 3014, 3016, 3017, 3021, 3021, 3023
installed components 3036, 3039, 3043
query feedback 3036
Roles 3026
This Helped 3036
contextual security
configuring 683
Contextual Security: Role Management Enhancements 2509
Contextual Security: Role Management Enhancements REST API 2509
Contextual Security
use access control rules 2526
contextual security manager 2456
Contextual Security: Role Management Enhancements 2510

© 2017 ServiceNow. All rights reserved. 4034
conversation
Legacy Notify
viewing 2965
conversions 1969
Core configuration 1233
Core Configuration
Form Administration 703
List Administration 672
overview 177, 964
Platform Performance 1995
time 993
corrupt index
regenerating 1101
create
bubble charts 380
create a custom table
create a table 1616
global default fields 1622
Create a Survey Wizard
configure the survey 658
create a dynamic effect 656, 656
create a survey page 655
define a survey question 654
test the survey wizard demo 657
create a UI policy
convert a UI policy to a data policy 758
define client scripts 757
on load check box 757
view-specific UI policy 757
Create an Advanced Wizard (Demo)
add a script 668
add field setter 668
create a panel 668
create fourth panel 668
create the wizard 666
create third panel 668
define a transition 669
define a variable 667
module in the asset contracts application 671
publish the wizard 671
test the wizard for a record with an asset 670
test the wizard for a record with no assets 670
create homepages
for specific users 486
global 486
new personal 485
personal based on existing 486
Create On-Call Schedules
enable on call notifications 1780
create tasks activity
use of with subflows 3554, 3751
creating
BSM map
icons 2206
indicators 2201
menu actions 2208
related items 2204
Legacy Notify
choice 2964
question 2963
response 2964
new
field 783
creating quizzes with forms
available data types 397
modify a published quiz 420
select a user for a category 412
set up a category 411
CSS styles
default 266
CSS theme
selecting 265
tHEME picker, enabling 264
CSS theme support
activating 264
CSV web service
file, posting 3380
parameters 3380
currency
conversions 960
export 961
forms 959
import 961
lists 958
properties 964
reference currency 956
reports 980
scripts 961
session currency 956
current security manager
determine 2456
custom 3992
custom filter URL
URL module 53
custom icon
Apple home screen 1178
custom processor
create 3877
custom tables
delete 1625
customizations
revert 1973
customize homepages
add items 491
change 493
custom homepage 493
delete 493
edit widget appearance 492
homepage 489, 490, 491
homepage layout 493
list gauge 491
modify 491
number of records 491
refresh 491
system administration 489
view 490
customized objects
during upgrades 1972

D

data archiving
affected records, verifying 1485
archive rule, activating 1482
archive rule, immediately activating 1482
archive rule, setting 1482
archive rule, tables and modules 1483
archive schedule, changing 1484
archived strings, language of 1485
creating rules 1480
plugin 1479
related records 1482
restored records, archiving 1488
restoring 1491
tables, querying 1484
viewing 1489
Data Dictionary Tables
system dictionary 1443, 1443, 1445
data import
schedule 1573
scripting options 1575
Data Lookup and Record Matching Support
activate the plugin 951
create a custom data lookup table 945
create a data lookup module 950
create custom data lookups 945
data lookup definition record 947
data lookup definitions fields 948
data lookup table 945
data lookup value 945
matcher field definitions fields 949
setter field definitions fields 950
troubleshooting 951
data management
database structure 1436
field normalization features 1439
import and export tools 1439
plugins 1438
scripts 1440
tools 1436
Data management
import export properties 1439
data policy
installed with 939
Data policy
converting to a UI policy 943
creating 941
creating from a UI policy 942
deleting all records 944
default sort sequence 939
field 939
data preserver
SAML certificates 1684
SAML properties 1681
SAML users 1684
data preservers
multiple provider single sign-on integration 1680
data source
choice list 1524, 1525
create 1526, 1841
Data Sources
Custom CSV files 1517
File 1512
File examples 1515
File retrieval methods 1513
FTP extended properties 1514
JDBC 1518
JDBC connection properties 1522
JDBC SQL statement requirements 1520
LDAP 1527
data types
survey questions 555, 584
database
audio files 204, 204, 205, 205
Database
storing images 205, 207, 208, 208
database graphs 2007
Database Rotation 1477, 1477, 1478, 1492, 1493
Date
validate 4020
date and time fields
classification 993, 994
personalizing the date format 994
personalizing the time format 994
Date and Time Fields 993, 996, 996, 996, 997, 997, 997, 997, 998, 998
daylight saving time
time zone 1053
user preferences 1053
debugging
session 3835
UI policies 3847
debugging classifications 3841
decision
classification 3841
decision matrix
viewing 445
viewing matrix 445
decision matrix
creating 438
decision matrices
administering 442
decrypted field value
return 2539
default sort sequence
list 680
defining
on-cancel script 3648
script on-cancel 3648
delegates related list
add to user profile 3800
debugging
delegation
active directory application mode (ADAM) 1877
ADAM 1877
delete
assessable record 350, 449
custom tables 1625
deleting all records
table 1623
demand management
assessment category
creating 324
Dependency Views
menus 2181
dialog
custom
displaying 3908
dictionary
spelling checking 859
dictionary attributes 1453, 1454
dictionary override
define 1444
dictionary overrides 1444
Did You Mean?
suggestions, configuring 1080
digest token authentication
enabling exit script 1948
disabling
attachment
table 749
drag and drop 749
feature 749
table
attachment 749
Discovery
script includes
using 3977, 3978
discovery performance metrics 2009
disk partition performance 2011
disposition
upgrade history 1969
document feeds
add show live feed button 2339
configure security 2339
disable 2340
installed components 2340
live feed ui action 2339, 2342
Document Feeds
live feed functionality 2343, 2343
Document ID Field 889
document scoring 1092
domain separation
activating and deactivating domains 2605
activating plugins 2582
adding to a domain field 2605, 2608
adding to a visibility domains list 2609
changing domain visibility 2618
configuration module 2606
contains domains 2587
creating contains relationships between domains 2609
creating domain-specific choice lists 2610
deactivating 2618
default domain scope 2620
delegated administration
example with domain specific applications and modules 2590
example with domain specific policies 2596
disabling domain numbering 2636
domain assignment 2583
domain query methods 2603
domain scope 2588
domain selection menus 2610
domain visibility 2585
domains and associated companies 2619
domains for delegated administration 2590
enabling domain logging and debug messages 2636
enabling the domain reference picker on UI11 and UI15 2611
enabling the simple domain picker on UI11 and UI15 2610
granting visibility domains to a group 2608
granting visibility domains to a user 2605
live feed 2353
making a domain the default 2622
manage MSP domains for records 2624
properties 2620
re-enable domain separation 2635
re-enabling delegated administration 2635
record value selection 2589
resetting all records to the global domain 2635
restricting access by role 2614
selecting a primary domain 2615, 2617
setup 2604
user preferences 2620
validating the domain tree 2632
viewing historical messages 2637
viewing MSP domain relationships 2615, 2637
viewing real-time messages 2636
viewing tables that use domain separation 2635
viewing the domain log 2633
Domain separation 2632
domain separation setup 2605, 2606, 2608, 2621, 2625, 2625, 2635, 2636, 2637
Domain separation setup 2605, 2606, 2608, 2621, 2625, 2625, 2635, 2636, 2637
dot-walking 138, 138, 140, 145, 149, 150
drag and drop
disabling 749
feature 749
DSN
global default 3482
duplicate email notifications
troubleshoot 838
Duration Field Value
setting 3812
DurationCalculator
relative duration 3882
simple duration 3881
dynamic approval forms 469
dynamic creation 888
E
e-signature approval
activating 464
approval tables, selecting 466
de-activate 464
local database, setting up 467
plugin, installed with 466

© 2017 ServiceNow. All rights reserved. 4037
SAML 2.0 authentication, setting up 467
SAML properties 464
setting up 467
Easy import
  Controlling the template row limit 1597
Easy Import
  add a record in the template 1595
data type validation 1601
import a record from the template 1597
import template 1594
paste data into the template 1598
template type validation 1600
template validation 1598
update a record in the template 1596
use 1594
e易
  Adding proxies 2662, 2662
attachment decryption job 2673
attachment encryption job 2671
attachments 2642
configure 2667, 2667
create encryption keys 2655
database recovery job 2675
decryption job 2672
dictionary attributes 2697
digital signature 2652
encryption application 2641
encryption job 2671
encryption key default properties 2658
encryption key management 2643
encryption proxy 2641
encryption rule actions 2678, 2678
encryption rule conditions 2677
encryption rules 2676
encryption types 2641
field encryption 2668
file store 2657
getting started 2640
installation 2644
installed components
  properties 2699
  roles 2700
tables 2699
instance configuration 2668
Java Cryptography Extension (JCE) installation 2652
JSON content 2679
key management 2643, 2670
key rotation 2670
key store management 2643
limitations 2639
Linux installation 2648
Linux uninstall 2669
logging 2676
mass encryption jobs 2670
mass key rotation job 2673
monitor 2675
NAE key store 2654
order preserving database properties 2660
order token repair job 2674
overview 2638
password encryption 2658
POST parameters 2678
prohibited keywords 2696
proxy configuration 2659
proxy configuration locked property 2659
proxy configuration properties 2651
proxy connection requirements 2647
proxy download 2648
proxy properties 2661
proxy system requirements 2645
proxy update 2669
secure SSL connection 2653
single key rotation job 2674
start the proxy 2667
stop the proxy 2668
target properties 2649
user account 2650
web proxy configuration properties 2651
Windows installation 2649
Windows uninstall 2670
XML content 2678
Edge Encryption environment
  memory 2646
  Number of proxy servers 2646
  redundancy 2646
Edge Encryption proxy
  error handling 2680
global methods 2695
editing
  Legacy notify properties 2973
  properties
    Legacy notify 2973
editing form
  edit multiple records in a list using 93
editing forms
  canceling changes 124
  inserting records 123
elements
  adding
    scrolling 3399
    scrolling
      adding 3399
email
  accounts
    creating 2722, 2723
    testing 2723
    alternate configurations 2705
    attachment limit properties 2753
    client
      display additional information 2861
      SMS 2865
client interface 2858
configuration 2701
configuring email client auto-complete 2860
controlling access to the client 2862
creating an email client template 2862
customizing the client 2859
displaying an editable form field 2864
displaying the Reply To field 2865
eexisting users, matching 2839

© 2017 ServiceNow. All rights reserved. 4038
filters
activating 2852
creating 2853
inbound actions
importing users 2839
inbound email
action criteria 2823
inbound email action processing order
activating 2851
configuring 2848
managing multiple filters in an inbound email action 2850
inbound email actions
creating 2832
example: creating a change request 2846
example: handling email replies 2844
example: logging a problem 2845
examples 2844
incoming mail server properties 2746
next steps 2755
notifications
baseline 2784
creating a notification filter 2909
events
business rules 3001
creating an event 3002
global 3007
passing parameters 3008
reprocessing 3008
states 2998
forcing delivery 2908
subscription-based
accessing 2898
adding an unsubscribe link 2908
adding notifications 2899
creating a service provider 2901
filtering notifications with a schedule 2910
making mandatory 2907
using a business rule 2999
OAuth 2.0 2712
OAuth authentication
activating 2712
outgoing mail server properties 2746
quick messages 2866
removing the email icon 2868
service availability 2701
setting the from address 2864
size limits 2751
size restrictions 2701
standard configuration 2702
subject line 2830
system mailboxes 2857
templates
applying to a notification 2809
calendar integration 2818
convert to HTML 2809
creating 2806
overriding a template value 2813
the client 2857
watermark 2830
Email
additional properties 2737
enable auditing
table 2563
enable email
basic 2703
basic email 2703
encoded queries
operators 159
Encoded Query Strings
generate through filter 84
encrypted contexts
remove 2538
Encryption
support 2530, 2531, 2532, 2536, 2541, 2542, 2544
encryption context ID
set in script 2538
encryption demo plugin 2539
encryption proxy
adding proxies 2662, 2662
configuration 2659
configuration locked property 2659
connection requirements 2647
create encryption keys 2655
digital signature 2652
download 2648
encryption key default properties 2658
encryption key management 2643
file store 2657
Java Cryptography Extension (JCE) installation 2652
key management 2643
key store management 2643
Linux installation 2648
Linux uninstall 2669
NAE key store 2654
order preserving database properties 2660
password encryption 2658
properties 2661
proxy configuration properties 2651
secure SSL connection 2653
start the proxy 2667
stop the proxy 2668
system requirements 2645
target properties 2649
update 2669
web proxy configuration properties 2651
Windows installation 2649
Windows uninstall 2670
encryption scripting 2537
encryption support
FAQ 2539
endpoints
Twilio
configuring 2971
Enforce and test access controls
Scripted REST APIs 3232
enter submits form
change preference 720
entity generation 1021
escalation intervals 1016
escaping types
jelly 3944
Event
event management systems 1709
event notification 1710
event log 2996
event queue 2996
event registration 1011
event registry 1023, 2997
events
registered by live feed 2350
schedule 1016
example
incident
priority 1 3399
priority 1 incident 3399
Examples
end to end tutorial 3371, 3372, 3372, 3374, 3375, 3375, 3376, 3377
web services c sharp .net 3371, 3372, 3372, 3374, 3375, 3376, 3377
EXCEL
parameters 3379
web service 3379, 3379
execution
order 3645
execution plan
script 3891
execution plan scripts
limitations 3891
write 3891
existing notification message filter
edit 2910
existing notification message schedule
edit 2910
existing records
incoming emails, matching 2828
existing users
email, matching 2839
export
assessments 347
Export Data
available formats 1497
break up a large export 1509
enable export debug logging 1509
example 1496
excel export threshold 1500
export directly from the URL 1504
export form data 1501
export format processors 1498
export limits 1499, 1499
export list data 1502
export list fields 1503
filter list result 1504
parameters 1506
properties 1499
scheduled reports 1504
url export 1507
url query 1504, 1506
xml export 1496
Export public key certificate 1889
export scorecard
image 541
Export set
  Schedule an export 1612
Export sets
  Cancel an export 1614
  create an export definition 1610
  Create an export target 1611
  create new 1609
  easy export 1610
  export definition 1610
  export history fields 1614
  Export set fields 1610
  Export target 1611
  History 1614
  scheduled export
    scripting 1613
  Scheduled export 1612

Export Sets
  properties 1615

Exporting data
  Exporting currency fields to Excel 1504

Exporting Data
  export data 1494, 1496, 1497, 1498, 1499, 1500, 1501, 1502, 1503, 1504, 1504, 1506, 1507, 1509, 1509

extensions to Jelly syntax 3937, 3937, 3937, 3938, 3938, 3938, 3940, 3941, 3941, 3941, 3942, 3942, 3942, 3942, 3943, 3943, 3943, 3943
External Authentication (Single Sign-On - SSO) 1802, 1904, 1909, 1928, 1939, 1944
extra coverage 1787

F

failed login attempts
  managing 2489
  viewing 2490
feature
  disabling 749
  drag and drop 749
fetch XML
  use code 3378
field
  add spell checking 858
  creating
    new 783
  spell checking 858
Field administration
  normalize 895, 896, 897, 898, 901, 903, 904, 906, 907, 910, 912
Field Administration
  choice list width 848
  choice lists
    change none display value 845
    customize 841
    define option 843
    delete 846
    field type 847
    remove none option 844
    reuse 843
    security 842
  create a wikitext field 839
  data lookup and record matching support 945, 945, 945, 945, 947, 948, 949, 950, 950, 951, 951
default choice lists
  integer values 849
  extend functionality of a wikitext field 840
field normalization 890, 891, 891
field normalization rules
  expressions 935, 937
  patterns 935, 937
field status indicators
  change colors 106, 789
Field Styles
  defining 851, 852, 853
field transformations
  create normalized queries 913
  create transformation records 914
data jobs 913
  identification 922
  run data jobs 920, 922
  scripting 913
  select transform types 915
test 918
  testing mode 914
field type
  behavior 894
  create 894
field types
  scripting 3810
field value
  null 3812
field values
  business rules and client scripts for controlling 3811
  comparable 282
  comparing 280
  dates, comparing 282
  dates, evaluating 282, 283
  operators and operands 281
Field Watcher 3842, 3842, 3843, 3844, 3845, 3845
fields
  HTML fields
    HTML sanitizer 2548, 2550, 2551, 2551, 2552
Fields
  creating new 784, 784, 786, 786
introduction 781
fields on a form 105
file
  attachment 750
  role 750
file size
  attachment limiting 750
filter out
  hidden groups 1754
filter potential assignees
  based on skills 1801
filters
  operators 159
  restricting 691
  suppressing 691
Filters
  create 79
  delete 78
  dynamic filter option 80
  edit 78, 79
  restricting 691
  save 78
  use 78
fixed currency 980
fixed price fields 957
Follow REST API conventions
  Scripted REST APIs 3231
form administration
  create a UI policy 753, 757, 758
  create a UI policy example 756
  form design 707, 707, 713, 713, 713, 714, 714, 714
form annotations
  add 716
  disable 719
  language support 718
  toggle 716
  types 719
form context menu 103
form design
  add a formatter 714
  add annotation 714
  display field 713
  form content 713
  new field 713
  new section 714
form fields 105
form pane tabs 124
form personalization
  activate 715
  administer 715, 715
  disabling 716
  manage 715
  role requirements 715
  user preferences 715
form section
  create 724
  delete 725
  hide 725
  show 725
form sections 723
form splits
  add property to enable 722
  enable 720
form submission
  double-check 2458
form templates
  automatically applied 725
  create from form 727
  save form as template 731
  saving forms as templates in UI11 730
  scheduling 733
  template bar 726
formatter
  approval summarizer 461, 762
  create 760
  create UI macro 759
  override with macros 761
forms
  , apply a template 123
  , editing 121
  activity formatter
    live feed 766
  add activity formatter 766
  adding charts 705
  adding related lists 705
  annotate 716
  canceling changes 124
checklists
  activating 132
  create from templates 133
  creating new 132
  deleting 135
  deleting templates 135
  enable 130
configure 703
configure customer updates indicator 706
configure insert for task records 705
configuring related list fields 118
control the label type 748
customize activities 767
define required fields 747
display information from other records 747
editing 123, 124
field status indicators 106, 788
form headers by version 98
form headers, UI11 101
form headers, UI15 100
form headers, UI16 98
form pane tabs 124
inserting records 123
personalize 715, 715, 715, 715
personalizing 125, 126, 716
require fields 746
sections 108
frequency 1104
function 3849
G

g_chat
  available methods 2310
properties 2439
security administrator role 2444
history access
control 2567
history calendar 2571
history list 2568
history sets
comparing CI snapshots 2577
differences with audit 2565
export historical CI snapshot 2577
homepage
adding 482
controlling access 482
defining tables for 482
Top Searches 482, 489, 490, 491, 491, 491, 492, 493
update top search statistics 483
using 482
viewing user preferences for time period 483
homepage administration
add to an update set and application 477
restrict content additions 476
secure a homepage 477
user preferences 477
homepage caching
configure properties 499
notes and limitations 500
homepage splash page
activate plugin 501
configure 501
homepage widgets
UI page reference 483
homepages
customize 489, 489, 490, 491, 491, 491, 492, 493, 493, 3798, 3799
delete 3799
gauges 478
render time 511
troubleshooting 513
Homepages
gauges 478, 478, 479, 479, 480
how record feeds work 2385
how ServiceNow creates a VPN 2548
HR case
view progress 3798
HR portal
submit request 3797
HTML fields
adding
media 806
administering 808
embedding
images 806
htmlArea HTML editor 811
htmlArea HTML toolbar 811
image uploads 815
images
embedding 806
linking
website 807
media
adding 806
TinyMCE deprecated tags 810
TinyMCE HTML editor
installed components
business rules 814
properties 812
TinyMCE HTML toolbar 809
website
embedding 807
htmlArea HTML editor 811
HTTP 500 error 1414
HTTP connection management
properties 3469, 3469
HTTP Response Codes 3218
human resources
HR portal 3795
submit from HR catalog 3797
I
I18N 2164, 2165
icon
attachment
configuring 751
configuring
attachment 751
icons
BSM map
creating 2206
modifying 2206
IdentifyReconcile API
POST 3211
identity provider
Active Directory Federation Services 1930, 1930, 1933
1933, 1933, 1938, 1938, 1938
IdP 1930, 1930, 1933, 1933, 1933, 1938, 1938, 1939
See also identity provider
image fields
using 815
images
embedding
HTML fields 806
HTML fields
embedding 806
immediately invoked function expression 3849
Impersonate a User
force logout 1762
invoke impersonate option 1761
log impersonations 1762
modify impersonate option 1761
on a mobile phone 1761
useful logins 1759
import
assessments 347
import data using import sets
cancel an import in progress 1571
control import table dictionary entries 1572
ignore bad data rows in CSV files 1577
Java example 1579
Perl example 1578
post CSV or Excel files directly to an import set 1576
preserve leading spaces in excel imports 1577
review the import set 1572
run import 1571
set the import log level 1573
import log
import run history 1580
Import set
Coalesce
Empty fields 1544
Import Set API
GET 3137
POST 3135
Import Set Performance
troubleshooting 1601, 1602, 1602, 1603, 1603, 1603, 1603
Import Set Scheduled Cleanup 1581
import set tables
web services 1691
import sets
performance improvements 2005
import sets
maximum row size 1551
Import Sets
Coalesce options 1542
Conditional coalesce
Example scripts 1542
easy import 1594, 1594, 1595, 1596, 1597, 1598, 1598, 1600, 1601
terminology 1511
web proxy 1433, 1433, 1433, 1434, 1435, 1435
importing
XML records 1604
Importing
ServiceNow instance 1604, 1605, 1606, 1606, 1606, 1607
XML records
matching existing records 1604
inactivity monitors
setting 1016
inbound email
action type 2827
recipients, processing 2841
setting field values 2842
sys_email 2842
troubleshooting 2844
inbound email actions
locked out user 2838
triggering 2838
Inbound Email Actions
example 2847
inbound event
integrate 2842
inbound JSON request
create notification 2842
Inbound REST
basic auth 3214
import set examples 3140
required roles 3215
resources 3055
response codes 3218
security 3214
supported headers 3217
URI version 3214
inbound SOAP
insert related records 3299
Inbound web services
examples 3354
incident
example
priority 1 3399
priority 1
example 3399
Incident Alert
Notify number groups 2911
incoming email
matching senders email address 2827
matching to existing records 2828
user creation 2839
values 2848
index
field, removing 1101
HTML 1096
removing 1101
indexing 1098
indicators
BSM map
creating 2201
modifying 2201
input variables
parallel flow launcher 3744
installation exits 2485
installation settings 3878
installed with
Content Management business rules 2050
Content Management client script 2050
Content Management roles 2049
Content Management script includes 2049
Content Management tables 2048
installed with live feed 2343
Instance
translate 987, 987, 988, 988, 988, 989, 989, 990, 990, 991, 991, 992, 992
instance locale
set 965
Instant Messaging 2330, 2330, 2331
integrate 1684
integration
data and process 1686
login 1686
Integrations
available 1687
Interface
using 28
Internal CA, generate certificate 1887
internationalization
support 965
Introduction to Fields
change 787
character counter 781
database field types 782
delete 788
delete multiple 788
field label 787
hint 787
status indicator 106, 789
  text field 781
types 777
IP Range Based Authentication
  access control 2480
  block a particular range 2480
  block everyone except a particular range 2480
IsJavaObject 3964

J
Java Apache Axis2 Web Services Client Examples 3354, 3355, 3356, 3357
Java digest algorithm
  encryption 1951, 2541
JavaScript debug window
  using 3848
JavaScript log 3847
JavaScript Log 3847, 3847
JavaScript regular expressions
  Java syntax 3814
javascript tools
  script includes 3963
JDBC
  SQL Server Integration 1521
JDBC driver
  choice list, edit 1524, 1525
  data source, create new 1526
  DB2 1525
  Install on a MID server 1523
  Sybase 1525
  unsupported data sources, adding 1523
JDBCProbe 1714, 1714, 1714, 1716, 1717, 1720
jelly replacement
  add escaping 3944
job
  schedule 1021
journal fields
  configuring
    suggested text 954
JsonNode
  encodedQueryFor(String tableName) 2685
  getAsString(String propertyName) 2684
  getIterator(String xPath) 2683
  getName() 2684
  iterator() 2683
  valueFor(String tableName, String fieldName) 2685
JsonNode API 2683
JsonNodeIterator
  hasNext() 2686
  next() 2687
JsonNodeIterator API 2686
JSValidator 3965

K
keyboard navigation
  use for list editing 90
keystore
  validate 2454
keyword search 1104
knowledge
  incident deflection and resolution 3026
  mobile access 2166
  view on mobile 2167
Knowledge
  Administration 2153
  Article view properties 2160
  Assign a manager 2153
  Category 2140
  create knowledge 2138
  Define a category 2151
  featured content 2143
  feedback 2137
  From incident or problem 2141
  Import a document 2141
  Incident Deflection and Resolution 3026, 3030, 3031, 3033
  Knowledge Base form 2153
  Knowledge homepage properties 2162
  Knowledge search properties 2158
  Manager 2143
  migrating access controls 2172
  mobile search 2168
  Move an article 2140
  Other knowledge properties 2162
  pin an article 2143
  portal properties 2155
  properties 2155
  Publish and retire workflows 2154
  Retired knowledge articles 2141
  search changes 2171
  searching knowledge 2131
  Supported import elements 2141
  user criteria 2143
  Workflows 2154
Knowledge management
  Request a knowledge base 2138
  Service catalog 2138
Knowledge Management
  homepage 2129
  internationalization 2164, 2165
  roles 2128
  set up 2124
  submissions 2172
Knowledge submissions 2172
Knowledge v3
  after migration 2170
  default knowledge base 2170
  knowledge article view 2171
  knowledge portal changes 2172
  migrate knowledge content 2175
  migrating functionality 2173
  migration guide 2169
  publish and retire workflows 2174, 2174
  replace modules 2173
  security changes 2171
  using legacy knowledge 2170

L
language internationalization support
  Choices table 971
configure a language as reading from right to left 968
Field Label table 970
global language 966
language table 967
Message table 969
service catalog buttons 976
show translated strings 973
Translated Name / Field table 968
Translated text 972
translation tables 967
user specific language 966
language plugin
activate 977
language support
form annotations 718
layout
add content 497
choose 496
concepts 494
custom 495
defining 494
LDAP
Active Directory 1848
assign field values 1848
authentication 1812
automatic operational status update 1838
automatic server validations 1835
binary data through mid server 1856
BLOB data 1856
certificates 2449
common scripts 1848
communication channels 1821
communication with MID server 1853
configuration options 1814
connection 1838, 1855, 1881
connection monitoring 1856
connection timeout 1838
create server 1823
data population 1812
data transformation 1844
deactivate 1847
default mapping 1847
exclude users 1848
extraction 1870
filters 1869
import 1847
import maps 1845
inactive accounts 1871
inactive users
disable 1871
integration
error codes 1860
troubleshooting 1858
integration via MID Server 1850
legacy import map 1846
listener
setting up 1831
login 1813
mapping relationships 1846
MID Server
troubleshooting 1856, 1859
monitor 1815
multiple domains 1818
notifications 1856
organizational unit 1838
OU 1838
persistent query 1814
persistent search 1814
query limits 1814
redundant server
creating 1827
creating from an LDAP server record 1828
refresh 1849
scheduled data imports 1845, 1856
schema 1890
script 1848, 1848, 1848
script examples 1873
secure connections 1814
server fields 1824
setting connection properties 1833
setup 1821, 1830, 1850, 1851, 1851
specifying attributes 1832, 1851
SSL 1830
SSL certificate 1823
supported servers 1813
testing connections automatically 1859
testing connections manually 1837, 1860
transform map 1842, 1846, 1849
transform script 1846
user account
deactivate 1872
disabled 1872
user auto provisioning 1845
verify mapping 1849
X.509 certificate 1823
LDAP Connection Failed 1856
LDAP integration 1812
LDAP Monitor
view 1816
LDAP records
inactive 1868
synchronize 1868
LDAP transform
choice action 1849
create 1849
ignore 1849
record creation options 1849
reject 1849
LDAPS
ADAM 1879
certificate, assigning 1880
connection 1881
public key certificate 1880
self-signed certificate 1879
left-pad numbers
in custom tables 856
legacy business service management map 2194
legacy chat
installed components
business rules 2305
properties 2302
tables 2303
use legacy chat 2311
legacy chat migration 2297
Legacy notify properties editing 2973 viewing 2973 public conference calls 2968 Twilio configuring 2969
Legacy Notify account status messages 2974 administering 2968 call conference 2966 conference call 2966 Conference call participants 2975 conversation viewing 2965 viewing conversation 2965
legacy survey migrate 601 legacy surveys migrate 602 limit the number of results use filters 3377 limited tablet UI support 1179 limiting attachment file size 750 file size attachment 750 Linux 2662 Linux stats 2012 list calculations 673 list control 679 list control settings for list editor 682 list controls 674, 678 list editor administering 681 contextual security, configuring 683 control settings, configuring 682 edit multiple records in a list using 92 personalize 92 use 90 user preferences, managing 683 list editor properties configure editable field types 681 disable list editing 681 list mechanic access, controlling 688 activating 688 List of Plugins 1234 list title menu 58 lists detail rows enabling 688 editing 90 personalizing 95 search 85 search by column 86 sort 88 switch views 70 lists, embedded 108 live feed administer 2338 bookmarking feeds 2411 bookmarking hashtags 2411 disable record feed 2339, 2342 document feeds 2338, 2339, 2342 domain separation 2353 following feeds 2411 hashtag security 2356 installed with 2343 limit access by role 2355 managing message content 2355 searches 2399 security 2353 selecting feeds 2409 table access rules 2353 table notification related feeds 2369 workaround posted 2366 table notifications non-task example 2368 problem resolved example 2366 task table notifications 2358 team security 2355 user preferences 2343 live Feed 2336 Live Feed activate 2338 added to your homepage 2412 browser support 2372 following other users 2407 messages 2404 multiple records 2386 post a message 2424 post content in live feed post a message 2424 records 2379, 2383, 2385, 2386, 2387 UI usage 2402 update profile 2402 using 2370 using groups 2388, 2389, 2389, 2390, 2390, 2390, 2391, 2391, 2392 using social IT post content in live feed post content in live feed using with UI 2414 viewing another user 2405 live feed teams security 2355 locales define 964 localization currency 986 form annotations 718 system 984 Localization instance 987, 987, 988, 989, 989, 990, 991, 991, 992, 992 localization properties 984 location setup 987
locations define 202
locked out user
  inbound email actions 2838
logging in 10
logging performance metrics 2014
Login
  finding denied IP addresses 2481
  IP range based authentication 2479, 2480, 2480, 2480, 2481
login landing page
  specifying 2478
Login Scenarios 2488
login security
  activating ServiceNow access control 2496
  configuring ServiceNow access control 2497
  ServiceNow access control 2494
login security settings 2478
logout button
  remove 2479
logout confirmation prompt
  enable 2479
Long-Running SOAP Request Support 3305

M
manager notifications
  enabling 368
managing
  tablet properties 1231
Managing Data
  system clone 1669, 1671, 1671, 1672, 1674, 1675, 1676, 1676, 1678, 1679
Manual Approvals
  Conditions 3700
  Input Variables 3700
  Result Values 3700
  States 3700
Many to Many Task Relations
  define a relationship type 1663
  define a task relationship allowed 1664
  mark as solution button 1667
  modify the displayed field 1666
  plugin manifest 1662
  request the plugin 1662
  task relationship type record 1664
  ui actions 1667
  view task relations 1665
many-to-many relationship
  creating 1623
  referencing 1623
map
  BSM
    administering 2200
map pages
  create 246
  create modules 248, 250, 251
  modules 248
  scripting 252, 253, 253, 254, 254, 255
Maps 2195
master
  template
    building 2065
match relevance by
  field, controlling 1093
matrix
  decision
    viewing 445
  viewing
    decision 445
maximum number of rows returned
  specify 3501
media
  adding
    HTML fields 806
    HTML fields
      adding 806
menu actions 2206
menu actions
  BSM map
    creating 2208
    modifying 2208
menu categories
  adding application menus to 46
menu category
  defining 45, 46
menu lists
  modifying 44
message
  Legacy Notify
    viewing 2964
metric definitions 1039
metric result
  field names 528
Metric Results 372
metric types
  create 338
Metric Types 348, 449
Microsoft
  .NET 3360, 3361
  web services client
    examples 3360, 3361
Microsoft Active Directory, configure SSL access 1886
Microsoft SCCM
  Integration 1695
  integration 2007 1696
  Integration 2012 1696
  Integration 3.0 1697
MID Server
  configure 2667, 2667
  integrate Edge Encryption 2667, 2667
mid server login credentials
  encrypting 2536
Migrate Knowledge
  after migration 2170
  default knowledge base 2170
  guide 2169
  knowledge article view 2171
  knowledge portal changes 2172
  migrate knowledge content 2175
  migrating functionality 2173
publish and retire workflows 2174, 2174
replace modules 2173
security changes 2171
using legacy knowledge 2170
migrate surveys 602
migrated survey components 603
migrated survey questions
review 604
mobile
  Apple and Android 1120
mobile app
    configure Connect 1215
    supported devices 1117
mobile app UI
    blur 1215
mobile banner
    define 1177
mobile browser interface
    back navigation 1230
    supported devices 1159
    UI actions 1230
mobile browsers
    module availability 1177
Mobile Client GlideForm (g form) scripting 3922, 3922
mobile configuration
    list search fields 1211
mobile device
    script testing 1179
    supported features 1117
mobile experience
    setup for administrators 1201
mobile interface
    access 1117
    home page
      favorite 1167
    mobile app 1117
mobile UI
    activity streams 1130
    application navigator 1126
    barcode scanning 1155
    condition sets 1135
    Connect Chat
      conversations 1150
    device location 1152
    device, define 1178
    disable 1179
    favorites 1124
    filters 1134
    form configuration 1207
    get started for end users 1115
    home page 1122
    list 1167
    lists 1128
    mobile app 1117
    service catalog
      catalog item 1141
      categories 1138
      shopping cart 1144
    smartphone interface 1158
    visualizations 1125
    web 1158
mobile UI, legacy
    configuration 1176
mobile web interface
    activate 1202
    disable 1230
modify
    schedule entry 1787
    modify form focus 719
modify transaction cancellation page 2038
modifying
  BSM map
    icons 2206
    indicators 2201
    menu actions 2208
    related items 2204
module icons
    create new 43
    modify 43
module link type 49
most active tags
    activating 176
    deactivating 177
    removing 177
most recent tags
    activating 176
    deactivating 177
    removing 177
multi-row gliderecord object
  forward 3569
multi-table processor
  creating 3876
multi-tenancy 2580
multiple approvers 468
multiple assessable records
  multiple category users, associating 354, 356
  stakeholders, deleting 355
multiple currency 980
multiple fields
  sort 680
  multiple price fields 957
Multiple Provider
  single sign-on
    using with ESS 1903
multiple provider single sign-on integration
  data preservers 1880
  system clone 1880
multiple transforms
  single import set 1529
Mutual authentication
  Outbound web services 3468, 3468, 3468
  Protocol profile 3468
MySQL overview 2015

N

Navigation Actions
  adding sites 183
  System Properties 185
  use cases 185
  users 184
navigation and UI
  context menu 58, 287, 288, 289, 291, 294, 294, 295
context menus 287
list features 57
Navigation and UI
view management 234, 236, 236, 237, 238, 238
view rule 239
Navigation Stack 177
network layout
basic email 2703
new customizations
translate 974
new department
adding 1749
new records
create 91
news
adding scroller 3401
scroller
adding 3401
news panel
adding
scrolling 3400, 3401
scrolling
adding 3400, 3401
node metrics 2017
Non-Interactive Sessions 1750, 1750, 1750, 1751, 1751, 1752
Nonce
implementing 2481, 2482, 2482
notification
calendar integration 2818
Notification Activities
Create Event 3719
Notification Activity 3720
notification message
subscribe 2878
notifications
deactivate for surveys 599
e-mail client
configuring email client auto-complete 2860
controlling access to 2862
creating an email client template 2862
customizing 2859
displaying an editable form field 2864
displaying the Reply To field 2865
interface 2858
quick messages 2866
removing the email icon 2868
setting the from address 2864
SMS 2865
message body limit properties 2752
previewing 2771
push
activating 2873
architecture 2870
business rules 2875
components 2875
creating a content record 2884
creating a push action 2884
creating a push message 2877
creating a push notification 2878
creating an attribute definition 2886
creating an attribute value 2888
developer set up 2876, 2880
example 2889
feedback 2872
handling failed messages 2895
installed with 2874
properties 2896
push application 2880
responses 2871
REST messages 2875
roles 2875
setting up devices on user preferences 2895
setup with a custom mobile app 2872
setup with the ServiceNow mobile app 2872
tables 2874
setting up devices on user preferences 2896
Notify
activate 2960
API 2931
call(String notifyPhoneNumber, String toPhoneNumber, GlideRecord conferenceCall) 2932
call(String notifyPhoneNumber, String toPhoneNumber) 2932
calls 2912
Client example 2960
Conference call 2913
conferenceCall() 2937
Configure Twilio 2910
Create a number group 2911
Creating Notify activities 2923
dequeueCall(GlideRecord callRecord) 2939
event handlers 2923
events 2923
Forward call activity 2917, 3711
forwardCall(GlideRecord call, String destination, String dtmf) 2940
getAvailableClients(String notifyNumber) 2937
getChildCallIDs(GlideRecord callRecord) 2934
getParentCallID(GlideRecord callRecord) 2935
getPhoneNumbers() 2931
getTokens() 2936
getTokens(GlideRecord callRecord) 2936
Hang up activity 2919, 3713
incoming calls
processing 2912
Input activity 2918, 3711
kick a participant 2913
kick(GlideRecord user) 2939
Languages 2924
migration 2961
mute a participant 2913
Notify on task 2924
Notify with incident alert 2925, 2925
Notify with on-call scheduling 2929, 2929
Number groups 2911
On-call: assign by acknowledgement workflow 2930
outbound requirements 2911
Play activity 2920, 3713
queueCall(GlideRecord callRecord) 2938
Record activity 2920, 3714
Reject activity 2921, 3714
Say activity 2921, 3715

© 2017 ServiceNow. All rights reserved. 4051
sendSMS(NotifyPhoneNumber notifyPhoneNumber,
String toPhoneNumber, String messageBody,
GlideRecord source)

sendSMS(String phoneNumber, String smsBody,
GlideRecord source)

sendSMS(String phoneNumber, String smsBody)

Twilio service

using SMS

variable substitution

WebRTC Client event handlers

workflow activities

Notify on task

Conference call from task

SMS from task

Notify with Incident Alert Management

Conference call

conference call information

launching

sending

SMS Notifications

viewing

Notify with on-call scheduling

NotifyNow key differences

Set up

NotifyAction

addConference() 2948
addDial() 2940
addGather() 2941
addHangUp() 2942
addPlay() 2943
addQueue() 2942
addReject() 2944
addSay() 2944
fromJson(String json) 2945

NotifyClient

SendDtmf(String digits) 2956

NotifyNow

addConferenceCallParticipant(String conferenceCall,
String participant) 2991
convertLocalPhoneNumberToE164(String userID,
String phoneNumber) 2992
getConferenceCallParticipants(String conferenceCallId,
Boolean isCallable) 2993
getFrequentlyCalledUsers(Number limit) 2994
getPreferredE164SMSNumber(GlideRecord user) 2995
getPreferredE164VoiceNumber(GlideRecord user) 2995
getPreferredEmailAddress(GlideRecord user) 2996
getReadyState() 2979
getStatus() 2980
initiateConferenceCall(String, String, GlideRecord,
Boolean) 2982
isCallable(String participant) 2983
isSMSCapable(String userI D) 2985
isVoiceCapable(String userI D) 2985
kick(GlideRecord participant) 2986
mute(GlideRecord participant) 2987
sendEmailQuestion(String emailAddress, String
question, GlideRecord sourceRecord, String
emailSubject) 2988

OAuth

activating 1955

APIs

scoped 1968

authorization code flow 1961

enhancements 1954

external client scenario 1956

profile parameters 1965

property 1955

provider 1961

revokeing a token 1968

scope 1961, 1964

setting up 1955

third-party provider scenario 1958

ODBC

behavior 3496

client applications 3499

data sources 3499

download and install the ODBC driver 3472

Excel 2010 3510

installation requirements 3472

set a user permission 3471

ODBC driver

configuring 3481, 3483, 3488, 3490

create a user 3471

Crystal Reports 2008 3516

define an ACL rule for the ODBC role 3471

download and install the ODBC driver 3472

ODBC support 3502

© 2017 ServiceNow. All rights reserved. 4052
supported software 3474
  test 3493
troubleshoot 3496

ODBC Driver
  Create the ODBC role 3471
download 3475
  grant required roles 3471
Querying table and column names 3496

ODBC driver quick start
getting started 3470

ODBC truststore 2665

OKTA SSO
  activating the plugin 1946
  instructing users to log in 1946
  setting up 1946

on-call schedule
  change the rota or span for a group 1782

On-Call schedule
  adjust an existing shift 1780
  create on-call schedule 1775
  management 1780
  substitute a shift 1780

on-call scheduling
  access roster for a rota 1779
  additional workflows for notify 1798
  create on call schedules 1775
  email workflow 1794
  escalation chain 1795
  escalation settings 1779
  escalation triggers 1792
  escalations 1791, 1794
  example escalation 1792
installed components
  business rules 1772
  client scripts 1772
  email notifications 1773
  events 1773
  properties 1770
  scheduled jobs 1774
  script includes 1771
  tables 1769
  user roles 1770
installed with 1769
  my group schedules 1783
  reminder lead time 1791
  resend reminders 1791
  rota escalation 1778
  rota reminder 1779
  rotate through rosters 1798
  trigger rule 1793
  trigger rules 1799
  upgrade 1798, 1798, 1798, 1798, 1799
  use notify 1797
  wizard 1798

On-Call Scheduling
  create on call schedules 1780

On-call scheduling with Notify
  Force communication channel 2930

one assessable record
  one category user, associating 356
onLoad 3848, 3848

OpenLDAP
  modify 1890
operators
  condition builders 159
  encoded queries 159
  filters 159
order
  execution 3645
order guide
  running automatically 3887, 3888, 3889, 3891

Order-preserving encryption 2647
organizational unit
  ADAM 1877
  create 1877
outbound email
troubleshooting 2844

Outbound REST
  authentication 3406
  Create a REST message 3404
  Define a header 3406
  Define an HTTP method 3405
  Generate a script preview 3411
  generate variables 3411
  MID Server 3406
  mutual authentication 3410
  OAuth 2.0
    Google Contacts tutorial 3408
  REST message elements 3403
  Scripting 3411
  Testing HTTP methods 3405
  Variable substitution 3410
Outbound SOAP
  envelope variables 3439
  External XSD file 3443
Outbound SOAP web service
  Video tutorial 3435
Outbound SOAP Web Service
  basic authentication 3449
  configure soap with a proxy 3450
  connectivity details 3449
  demonstration video 3449
  mid server 3442
  mutual authentication 3450
  outbound soap security 3449
  soap message 3435, 3435
  soap message functions 3437
  soap messages 3442
  test the soap message 3440
  use soap message in a script 3441
  variable substitution 3439
  web service security 3449
Outbound web services
  Mutual authentication 3468, 3468, 3468
override execution plan
  script 3892

page templates 2066

© 2017 ServiceNow. All rights reserved. 4053
pages
  make private 2477
  make public 2477
panel flow
  basic 641
panel transition
  define 641
parallel flow launcher
  example use of 3748
  input variables 3744
  script values in 3748
  subflow activities 3744
  use of create task activity 3554, 3751
  WorkflowCoordinator object 3747
parallel homepage
  rendering 510
parallel processing
  configuring 1076
parameters
  condition 2210
  script 2211
parent record 728
password reset 2491, 2492
password reset message
  add 2493
password validation rules
  strengthen 2487
pause conditions 1016
PDF Web service
  parameters 3391
Percent Complete Field
  convert 821
  create 821
performance
  evaluating 2000
Performance Analytics
  API 3164, 3164
  REST API examples 3167
Performance Analytics API 3167
Performance Metrics 2007
Perl
  API change 3335
  client 3363
  examples 3363
  web services 3363
Perl API
  approval 3342
  architecture 3316
  classes 3318
  configuration 3339
  examples 3343
  install 3312
  RequestedItem 3338
  SC Task 3341
  system requirements 3312
  ticket 3341
Perl API attachment 3340
Perl API Dictionary 3338
Perl API GlideRecord 3329
Perl API Incident 3332
Perl API Request 3336
Perl API Task
  subroutines 3331
personal lists
  administering 687
  managing 688
personalizing lists 672, 672, 673, 673, 674, 679, 679
Personalizing Lists 678
pinning
  activity level in workflows 3650
  in workflows 3650
  workflow level 3650
Planned Task 1649, 1649, 1650, 1651, 1652, 1653
Planned Task Hierarchy 1654
platform 6
platform administration 295
Platform Performance
  browser settings 1999
  Stats Tools plugin 2000
plugin
  Enhanced Web Service Provider - Common 3293, 3294
Plugin
  Google custom search integration 1109, 1694
plugins
  activate 1233
  general 1233
  options for activating 1233
  purchase 1234
Plugins 1234
pop-up delay 867
pop-up information
  customize 866
POP3 server 2718, 2720
populate CMDB
  computer information 3799
post content in live feed
  add @mentions 2433
  add a knowledge article link 2431
  add a link 2430
  add a poll 2431
  attach files 2426
  clipboard 2428
  copy an image 2428
  delete a message 2434
  like a message 2434
  message 2431, 2433
  post 2426, 2430, 2431
  reply 2426, 2430, 2431
  reply to a message 2424
price fields 957
priority 1
  example
    incident 3399
    incident
      example 3399
Problem Management
  Perl API problem 3336
process flow formatter
  activating 773
  add 770
  attach to form 772
  create 770
processor
  multi-table 3876
processors
  API components 3875
  creating 3876
  fields 3874
  secure 3875
  simple 3876
Production System 1699
properties
  base-system 1338
  installed with live feed 2345
Legacy notify
  editing 2973
  viewing 2973
properties module
  creating 1413
Proxy database
  size 2647
proxy server
  set initial memory 2661
  set memory 2661
  set upper bound memory 2661
public key certificate
  exporting 1880
  LDAPS 1880
Public key certificate, export 1889
public pages module
  activate 2478
Publishing Wizards 645, 646, 647, 648
pull data
  local data store 3378
punctuation 1094
purchase plugins 1233, 1234
Python
  client 3368
  example 3368
  web services 3368

Q
queries 52
query
  test 3496
query method
  configuring 1076
query tables 3883
question
  Legacy Notify
    creating 2963
    viewing 2962
question data types 555, 584
questions
  in quizzes 413
questions for surveys 583
quiz
  questions 413
quiz designer 400, 403, 409, 410, 411, 421, 421, 424, 424, 425, 610, 613, 622, 623, 624, 624, 625, 625, 625, 625
Quiz Designer 400, 402, 421, 610, 612, 619, 626, 628
Quiz Overview module 399, 609
quiz reports 425, 635
quiz scorecards 386, 426, 426, 427, 429, 431, 433, 629, 631, 633
quizzes
  activate 399, 609
  administer 396, 608
  categories 398, 607
  category user 398, 607
  create 402, 612
  process 396, 608
  questions 398, 607
  roles 398, 608
  set up 396, 608
  templates 398, 607

R
range calculator scripts 1013
ranking
  stories 702
raw field
  create 938
  read only role 1729
  read-only fields
    pop-ups 867
received emails
  reprocess 4003
recently-viewed records
  history 176
record
  deleted 1891
  record feed 2385, 2386
  record feeds
    add live feed 2342, 2342
    configure 2341
    configure security 2342
    context menu 2342
    disable 2339, 2342
    form header 2342
record history
  view 2566
Record Numbering
  managing 853, 854, 856, 857
record producer 3028
records lists
  enable text searches 1097
  text searches 1081
recovery
  active directory application mode (ADAM) 1879
  ADAM 1879
redundancy
  active directory application mode (ADAM) 1879
  ADAM 1879
reference
  qualifiers 3898
Reference
  qualifiers 3985
reference autocompletes
  using wildcards 886
reference currency 956
Reference Field
  auto-complete 877
dictionary 877
reference fields
  add 861
dynamic creation 888
recent selections 889
Reference Fields
  auto-complete
  UI features 878
cascade delete rules 876
decorations 863
define reference key 887
display values 861, 862
lookup 863, 865
reference icon 865
reference styles 869
related incidents icon 868
select 862
show workflow icon 868
tree picker 865
reference glide list from a script 3861, 3861
reference icon customizing 866
reference qualifier set up 1756
reference qualifiers 869, 869, 870, 870, 871, 872, 873, 874, 875, 875
registering
  event 1011
related items
  BSM map
    creating 2204
    modifying 2204
related list
  select 694
related lists
  adding fields 118
  configure the edit option 112
  configure when a related list loads 111
  create a default filter 111
  create a new related record 111
  create default filter 120
  create defined 113
  example of defined 114, 117
  incidents by same caller 114
  select related records 110
related record 728
relative duration define 1006
use 1007
Remember Me checkbox
  default value, changing 2490
  removing 2490
Remember me login cookie 2457
reminder 1655
replication metrics 2007
report lowest table
  Glide hierarchy 3601
request
  getAsJsonContent() 2681
  getAsXmlContent() 2682
  getEncodedQueryFor(String tableName) 2680
  valueFor(String tableName, String fieldName) 2680
request API 2680
request plugins 1233
required form fields
  plugin, activate 746
reset password 2492
resolve conflicts 1971, 1972
response
  Legacy Notify choice 2964
  creating 2964
Response time indicator 165
REST 3135, 3406
REST API
  Aggregate 3119
  aggregate examples 3126
  CORS domain requirements 3216
  CORS support 3215
  define a CORS rule 3216
  Dot-walking 3219
  explore a table 3055
  explorer 3055
  explorer interface 3056
  getting started 3045
  Performance Analytics 3164, 3164
  Performance analytics examples 3167
  Performance Analytics security 3167
  user role inheritance 3210
REST API explorer
  interface 3056
REST message
  Basic auth profile 3407
  Define a header 3406
  elements 3403
  REST message with basic auth 3407
  REST message with OAuth 3408
  Testing HTTP methods 3405
  Variable substitution 3410
REST Message
  Define an HTTP method 3405
  Generate a script preview 3411
  Scripting 3411
REST OAuth
  Google provider tutorial 3408, 3409, 3410
RESTAPIRequest
  body 3233
  getHeader(String header) 3235
  getSupportedResponseContentTypes() 3235
  headers 3235
  pathParams 3233
  queryParams 3233
  queryString 3234
  url 3234
  url 3234
RESTAPIRequestBody
  data 3237
  dataStream 3237
  dataString 3238
  hasNext() 3238
  nextEntry() 3239
RESTAPIResponse
  getStreamWriter() 3240
  setBody(Object body) 3240

© 2017 ServiceNow. All rights reserved. 4056
setContentType(String contentType) 3243
setError(Object error) 3243
setHeader(String header, String value) 3242
setHeaders(Object headers) 3241
setLocation(String location) 3241
setStatus(Number status) 3242

GET APIResponseStream
writeStream(Object stream) 3244
writeString(String data) 3244

GET Messagev2
execute() 3413, 3413
GET MessageV2() 3412
GET MessageV2(String name, String methodName) 3412
setHttpMethod(String method) 3418

GET MessageV2
asynchronous example 3433
Example direct message 3432
Example empty message 3433
Example with MID server 3434
getEndpoint() 3414
getRequestBody() 3414
getRequestBody(String headerName) 3415
getRequestBodyHeaders() 3415
saveResponseBodyAsAttachment(String tableName, String fileName, String encryptContext) 3423, 3424
setAuthenticationProfile(String type, String profileId) 3425
setBasicAuth(String userName, String userPass) 3416
setEccCorrelator(String correlator) 3416
setEccParameter(String profileName) 3419
setLocation(String location) 3417
setEndpoint(String endpoint) 3417
setHttpTimeout(Number seconds) 3418
setMIDServer(String midServer) 3418
setMutualAuth(String profileName) 3419
setQueryParameter(String name, String value) 3420
setRequestBody(String body) 3421
setRequestBodyFromAttachment 3425
setRequestHeader(String name, String value) 3421
setStringParameter(String name, String value) 3421
setStringParameterNoEscape(String name, String value) 3422

restore deleted records
configuration records 2579
configure 2579
data 1626
data records 2578
deletion audit 2579
deletion audits 1626, 2578
restore 2578
system table 2579

restoring deleted records and references 1626
restoring system tables 1627

GET ResponseV2
getBody() 3427
getCookies() 3427
getErrorCode() 3428, 3428
getHeader(String name) 3429
getHeaders() 3429
getQueryString() 3430
getResponseAttachmentSysId() 3430
getStatusCode() 3431
haveError() 3431
waitForResponse(Number timeoutSecs) 3432

restrict a field
example 684
with a condition, example 686
with a script, example 685

restrict a table
example 683

retrieve displayed values
choice list 3930

Return informative HTTP status code
Scripted REST APIs 3231

Return useful error information
Scripted REST APIs 3232

revert customizations 1973
role
attachment 750

group
assign role 1723
creating 1723

security jump start 1738

Roles
base system roles 1724, 1728
rollback to
behavior 3764
Rollback To
States 3702

rollback to activity 3764
rota active 1787

RSS Feed Generator 3395, 3396, 3397, 3398
RSS Feed Reader 3398, 3401
rules 3850

S

SAML
2.0
Java keystore 1925
user provisioning 1942, 1942
concepts 1904
data preservers 1681, 1684, 1684
identity provider certificate 1923
logout endpoint 1938
install 1923
system clone 1681, 1684, 1684
SAML 1.1
migrating 1940
SAML 2.0
integration with other features 1930
troubleshooting 1943
Sample C 1951
schedule entry
modify 1787
Schedule Pages 3966, 3966, 3968, 3969
schedule script execution 1019
schedule time off 1787
scheduled data import

time zone 1056
scheduled job 1023
scheduled report 1017
scheduled reports
time zone 1056
Schedules
using 1040, 1040, 1042, 1044, 1045, 1045, 1047, 1048, 1051, 1052
scoped applications
ACLS 2506
scorecard
view 529
scorecards
averages 386
categories 387
category metrics 388
head to head compare 389
history 392
links
create 384
live feed 393
surveys
average ratings 534
category results 529
exporting as an image 600
history 536
question results 532
view 384
Scorecards
ratings 395
script
client 3902
on-cancel

defining 3648
parameters 2211
Script
client
helper functions 3917
script actions
installed with live feed 2347
script API
service catalog 3897
script include
AJAXClientTiming 3931
GenericException 3963
IsJavaObject 3964
JSValidator 3965
script includes
installed with live feed 2347
suppress breadcrumbs and filters 691
Script Includes
change privacy 3865, 3866
client-callable 3865, 3866
privacy settings 3865
script sandboxing 2458
script syntax error
check 3830
script testing
mobile device 1179
scriptable java objects 3569
Scriptable Service Catalog Variables 3894
Scripted REST API
core concepts 3222
streaming vs large objects example 3250
Scripted REST APIs
Access controls 3224
add a version 3229
Available error objects 3225
Build tests verify functionality 3232
current type 3230
create 3227
create a resource 3227
Define query parameters 3228
enable versioning 3229
Enforce and test access controls 3232
Error objects 3224
Examples 3245
Explore 3231
Follow REST API conventions 3231
Maximum request size 3230
Require resource ACL 3229
RESTAPIRequest 3232
RESTAPIRequestBody 3236
RESTAPIResponse 3239
RESTAPIResponseStream 3243
Return informative HTTP status code 3231
Return useful error information 3232
roles 3223
script samples 3245
security 3224
Sending error objects 3225
service headers 3228
streaming file attachments example 3254
URI format 3222
Use versioning 3231
versioning 3223
scripted web service 3350, 3351
scripted web services
installed with live feed 2348
scripting
API suggestions 3816
business rules 3862
messages 3956
Scripting
  business rules 3839
currency field 3895, 3895, 3895
  price field 3895, 3895, 3895
Some fields
  adding 3398
scrolling
  adding 3399
    elements 3400, 3401
    news panel
search
  attachments 1102
  phrase searches 1085
wildcards 1086
search filter
  remove 2401
search for errors
  by line 3832
search groups 1069
search order
  field level rule 1732
search phase 1086
search phases 1086
search process 1086
search results 1069
searching
  boolean operators 1083
searching
  Google 1109, 1109, 1109, 1110, 1111, 1113, 1114, 1114, 1164, 1164
section 508 compliance features
  changing the color of mandatory field status indicators 1431
    disable list editor 1431
    enable accessibility 1431, 1431
    search method tips 1432
    set up 1430
    use skip links 1432
security
  active directory application mode (ADAM) 1874
  activity formatter 764
  extended table security 2455
  live feed 2353
security options 2457
security rules
  installed with live feed 2353
Security Scripts
  useful 2473, 2473, 2477
self service password reset 2491
self-signed certificate
  ADAM 1879
  LDAPS 1879
Server API 3962
  Server API reference 3951
server scripting
  DurationCalculator 3878
Server scripting
  email notifications 2776
Server Scripting
  DurationCalculator 3881, 3882
    email notifications
      changing link text 2778
      enabling links 2778
      example scripts 2780
      linking to content pages 2779
      linking to related records 2779
      mail script API 2781
      using JavaScript 2777
    script include form 3863
    script includes 3862, 3864, 3865, 3865, 3866
    transform map scripts 1530, 1530, 1532, 1532, 1533, 1534, 1534
service administration
  wizard 643
service catalog
  client scripts 3907
Service Catalog
  access controls 2468, 2469, 2471, 2472, 2472
  Client Scripts 3906, 3907, 3908, 3908
  localizing price 961, 961, 978, 981, 982, 982, 982, 983, 983
service catalog checkout
  adding 3986
Service Catalog requests
  add new request items 3789
  delete request items 3792
  place a request 3793
  view request status 3798
service catalog variables
  reference qualifiers for filter names example 876
service delegation 3800, 3800, 3800
Service Desk 2330, 2330
  service level agreements
    timing 1040
ServiceNow
  intranet 1713, 1713, 1713, 1713
ServiceNow Access Control
  audit logging 2498
ServiceNow administration 295
servicenow functionality
  desktop required 1180
ServiceNow Perl API
  constructor 3320
  examples 3327
  subroutines 3320
ServiceNow platform 295
ServiceNow Platform 6
ServiceNow servlet 2019
ServiceNow Store Usage Overview 1766
session client data 3912
session currency 956

© 2017 ServiceNow. All rights reserved.  4059
set up
   mobile experience for admins 1201
set up VPN
   ServiceNow and business network 2544
Setting Up Chat Queues for Help Desk Chat
   add custom link 2335
   assign service desk staff 2333
   chat queue 2333
   define a chat queue 2332
   end user 2334
   ess portal 2335
   help desk chat 2334
   link syntax 2335
   monitor chat queues 2336
   monitor help desk chat tasks 2336
share surveys
   sending invitations 597
Share Surveys
   assign a survey to a survey user 597
sharing
   surveys 598
shortcuts
   keyboard 17
signatures for assessments 347
simple search
   perform 2399
simple security manager 2456
single logout
   Active Directory Federation Services 1938
single sign-on
   multi-provider
      activating 1893
      choosing the IdP 1902
      configuring users 1894, 1898
      IdPs 1898
      installed with 1892
      logging in 1901
      modifying the primary 1894, 1898, 1898
      selecting the default 1898
      setting up 1894
      testing connections 1898
      troubleshooting scripts 1901
      troubleshooting test connections 1901
Single sign-on
   digest token authentication
      creating links 1950
      entering multi-provider single sign-on properties 1949
      entering single sign-on properties 1948
      generating a digest token 1948
      sample implementation 1950
      testing the integration 1947, 1947, 1950
single-currency mode 956
single-row gliderecord object
   forward 3569
Site
   creating 2052
   sites
      create 2050
size limits
   joins 1416
skill
   create 1800
   skills management
      activate 1800
   skipped updates 1972
   slow query logs
      use 2002
slushbuckets
   arrange selections 151
   find available items 152
   items
      setting the number of visible in the available column 153
      use 151
smartphone
   application menus 1203
   help screen 1218
   help text 1217
   home pages 1214, 1215
   localized help 1223
   modules 1204
   online help 1216
   table title 1213
smartphone interface
   access 1161, 1161
   application navigator 1203
   blackberry 1159
   client scripts 1227
   define devices 1230
   home page
      favorite 1166
      recent record 1169
      tag 1170, 1171, 1173
      journal stream 1174
      related list 1173
      shortcut 1162, 1163
      supported features 1159
      UI actions 1229
      UI policies 1225
      unsupported features 1160
      URL redirect 1229
      use 1161
SMTP server 2716, 2720
SOAP
   Outbound web service 3434
   Retrieve a large number of records 3377
   security 3283
   web service 3255, 3273, 3276, 3282, 3282, 3298, 3298, 3298, 3299
SOAP default security policy 3296
SOAP message
   hierarchical 3303
SOAP requests, signed
   certificates required 3294
SOAP session
   management 3304
   reporting 3304
SOAP session properties 3304
SOAP session report
   view 3307
SOAP strategies 3377
SOAP web service
FAQ 3307
PERL script, example 3307
security 3307
SOAP message, example 3307

SOAP Web Services
scripted web services 3345, 3350
SOAPMessagev2
execute() 3450
executeAsync() 3451
SOAPMessageV2
Asynchronous message example 3461
Direct message example 3462
getEndpoint() 3452
getRequestBody() 3452
getRequestHeader(String headerName) 3452
getRequestHeaders() 3453
Recordless message example 3463
setBasicAuth(String userName, String userPass) 3453
setEccCorrelator(String eccCorrelator) 3454
setEccParameter(String name, String value) 3454
setHttpTimeout(Number timeoutMs) 3455
setMIDServer(String midServerName) 3456
setMutualAuth(String profileName) 3457
setRequestBody(String requestBody) 3457
setSOAPAction(String soapAction) 3458
setStringParameter(String name, String value) 3459
setStringParameterNoEscape(String name, String value) 3459
setWSSecurity(String keystoreId, String keystoreAlias, String keystorePassword, String certificateId) 3460
SOAPMessageV2() 3461
SOAPMessageV2(String soapMessage, String soapFunction) 3461
SOAPResponseV2
getBody() 3464
getCookies() 3467
geretCode() 3464
ger(errorMessage) 3464
getHeader(String name) 3465
getHeaders() 3465
getStatusCode() 3466
haveError() 3466
waitForResponse(Number timeoutSecs) 3467

social it administration
get started 2300
social it 2300
Social IT Administration
document feeds 2343, 2343
set up chat queues for help desk chat 2332, 2332, 2333, 2334, 2335, 2335, 2336, 2336
Social Q&A
Answer a question 2134
ask a question 2133
comment 2134
delete a question 2135
edit a question 2135
enable questions in a knowledge base 2133
Language support 2136
notifications 2135
question 2133
share a question 2136
subscribe to a question 2135
tags 2136
vote 2135
sort order
remember 680
special characters
XML file 1668
spell checking
add to a field 858
dictionary 859
field 858
SQL Server
2008 3503, 3503
2012 3503, 3503
Linked Server 3503
SSL
LDAP 1830
SSL access, configure Microsoft Active Directory 1886
stakeholders 352, 1733
stand-alone CA, set up 1887
standard browsers
module availability 1177
Standard Import Set Tables 1544
state change
form color 3991
state flows
field properties 521
installed components
business rules 514
script includes 515
tables 514
rebuilding 520
State Flows
using 516, 516, 519, 520, 520, 521, 521
static WSDL 3351, 3351, 3352
Static WSDL 3350, 3350, 3351, 3352
statistics
text indexing 1101
Stats Tools
performance evaluation 2000
status
text indexing 1101
stop words
automatic, configuring 1106
global, configuring 1105
table-specific, configuring 1105
stories
ranking 702
string fields
configuring
suggested text 953
style in Content Management
design theme 2107
Style in Content Management
create a new frame UI macro 2109
define frame in style sheet 2108
design themes 2106
frame 2108
style sheets 2105
subflow activities
create task activity and 3554, 3751
parallel flow launcher example 3748
parallel flow launcher for 3744
parallel flow launcher input variables 3744
script values in 3748
WorkflowCoordinator object in 3747
subflows
inputs for 3556
invoking in a parent workflow 3557
preparing for use in a parent workflow 3557
variables passed from parent workflow 3556
submit a case
human resources 3797
suggestion fields
adding 952
using 951
Support
encryption 2530, 2531, 2532, 2536, 2541, 2542
supported interfaces 1707
supported tablets 1179
survey
customize 569
Survey
groups 565, 565, 566, 566
users 565, 565, 566, 566
survey administration
survey definitions 575, 576
Survey Administration
share surveys 597
survey definitions
modifying 572
schedule periods 575
survey designer
configuring a survey in 546
elements of 542
survey management
categories 549
categories, creating 548
category weighting
survey designer 562
key terms 525
questions 583
roles 524, 541
scorecards
average ratings 534
category results 529
exporting as an image 600
history 536
question results 532
survey designer
categories for 547
configuring category weights in 562
configuring surveys to allow retakes 566
editing a survey with 562
elements of 542
publishing a survey in 567
questions for 549
selecting survey recipients in 564
survey results 525
template questions 576
Survey Management 522
survey properties
ingoing 568
survey question header
change color 568
survey question migration 602
survey questionnaires 606
survey questions
changing order 593
create or update 580
Survey questions
template 591
survey responses 527
survey results
view 527
surveys
appearance, customize 567
assigning to user 597
auto-notification 599
designing 542
question data types 555, 584
sharing 597, 598, 598
survey definitions 572
survey designer elements 542
taking 605
view instances 562
synchronize
LDAP records 1868
syntax editor
editing JavaScript scripts 3817
keyboard shortcuts 3819
navigate to a line number 3830
script macros 3834
syntax editor plugin 3835
sys audit record
information 2562
sys audit table
understanding 2562
sys_id 3365
sys_properties.list
property, adding 1411
syslog probe 1705
system clone
multiple provider single sign-on integration 1680
SAML integration 1681, 1681, 1684, 1684, 1941
System clone
cloned to an instance on a different version 1676
decline a table from cloning 1680
run post clone cleanup scripts 1674
start a clone 1672
System Clone
allow a clone 1675
cancel a clone 1675
create a clone target 1671
delete a data preserver 1679
preserve data 1678
send clone status notifications 1675
view active clones 1676
view clone history 1676
system diagnostics
running 2021
system dictionary
  default sort order 680
  modifying 1445, 1445
System Dictionary
  create dictionary entries 1443
  dictionary entries 1445
system localization 984
system log messages 2038
system logs
  restore deleted records 1625, 2578, 2578, 2578, 2579
system properties 1338
system property table
  property, adding 1411
System Scheduler 1024
System Usage 1766

T

tabbed forms
  display 746
table
  attachment
    disabling 749
    viewing 749
deleting 1625
disabling
  attachment 749
  extending 1417
table access
  live feed 2353
table administration
  activating the restore deleted records plugin 1622
  restoring deleted records and references 1626
  restoring system tables 1627
  unique record identifier 1633, 1633, 1633, 1633
table api
  curl examples 3107
Table API
  DELETE 3093
  explore a table 3055
  GET 3057
  GET id 3072
  JavaScript 3114
  PATCH 3086
  Perl 3099
  POST 3066
  PUT 3078
  python 3103
  server-side JavaScript 3106
  table api language examples 3098
Table API Ruby
  examples 3095, 3116
table auditing history 2564
Table extension
  example 2039
table field
  prevent auditing 2564
table hierarchy
  extension model 2028
table index 1634
table information
  not audited 2564
table notifications
  examples in live feed 2361
  setting up for non-task tables 2360
table rotation
  apply 2025
tables
  delete 1625
Tables
  schema map 1627, 1627, 1627
tables installed with live feed 2344
tables module 1630
Table & Mobile UI 1160
Table and Mobile UI
  smartphone interface 1176
  tablet interface 1180, 1180, 1181, 1182, 1184, 1185, 1186, 1187, 1189, 1190, 1191, 1191, 1193, 1196, 1196, 1197, 1198, 1198, 1199, 1199
tablet interface
  favorites menu 1183
tablet properties
  managing 1231
tablet support 1179
tablet UI
  administering 1231
tag
  Glide 3931
  Jelly 3931	
tags
  assigning 166, 166, 167, 167, 167, 168
  editing 168, 169, 169, 170, 170
Task Active State Management business rule 3861
Task Activities 3727, 3727, 3727, 3728, 3731, 3762
task parent breadcrumbs formatter
  customize 774
task table 1655
Task table
  creating a task 1635
  important Task table fields 1643
  journal fields 1647
  modifications 1656
  modify the task interceptor 1647
  Planned Task plugin 1635
time cards 1656, 1657, 1658, 1658, 1658, 1659
  tools for driving tasks 1660
Task Table
  assignment lookup rules example 1636
  assignment module rule 1639
  assignment rules 1637
  assignment rules module 1637
  baseline assignment rules example 1638
  business rules 1637
  condition editor example 1637
  data lookup rules 1637, 1642
  defining assignment rules 1636
  many to many task relations 1661, 1662, 1662, 1663, 1664, 1665, 1666, 1667, 1667
  script example 1638
  workflow assignments 1638
  task table hierarchy
  flattening 2028
task table notifications
live feed 2358
tasks
delegate 3800
out of box task fields 1648
tasks workflow 1656
template
apply from a module 733
building
master 2065
master
building 2065
scripting 733
templates
apply in a form 123
Test LDAPs connectivity 1888
text index 1098
text indexing
statistics 1101
status 1101
themes
company default 265
create 265
customize 265
Theming an Instance 257, 262, 263, 264, 271
third-party 1684
this ServiceNow instance 2030
threads
performance monitoring 2006
time
display 1008
scheduled jobs 1023
Time
time zones 1056, 1057
validate 4020
view system logs 1057, 1058, 1056, 1058, 1059, 1061,
1062, 1062, 1062, 1063, 1063, 1064, 1064, 1065,
1065, 1066, 2552, 2553, 2555, 2557, 2558, 2558,
2559, 2559, 2560, 2560, 2561, 2561, 2562, 2562,
2750, 2998
time worked fields 1040
time zone
choice list 1055
scheduled data import 1056
scheduled reports 1056
time zones
daylight saving time 1053
email notification 1053
scripting 1053
service level agreement 1053
user preferences 1053
Time-Related Functionality
introduction 1012
Timeline Pages 1024, 1030, 1033, 1034, 1035, 1037
Timeline Visualization
create 3770
create view 3774
use Timeline Visualization 3775
timeline visualizations
installed components 3787
Timeline Visualizations
lanes 3775, 3775
dcards 3775
personalize timeline visualizations 3776
settings pane 3779
slider and slider track 3782
view timeline visualizations 3780, 3783
work with timeline visualizations 3783
Timelines
using 1025, 1025, 1027, 1028, 1028, 1029
timing functionality 1039
TinyMCE HTML deprecated tags 810
TinyMCE HTML editor
installed components
business rules 814
properties 812
TinyMCE HTML toolbar 809
transaction
cancel timer 2003
cancelling 2003
Transaction quota properties 2033
Transaction quotas 2032
Transaction Quotas 2034, 2034
transform categories 933
transform category
create 934
transform definitions
create 923
transform map
create 1556
scripts 1844
Transform Map Scripts
time worked fields 1040
time zone
time zones 1056, 1057
validate 4020
view system logs 1057, 1058, 1058, 1059, 1061,
1062, 1062, 1062, 1063, 1063, 1064, 1064, 1065,
1065, 1066, 2552, 2553, 2555, 2557, 2558, 2558,
2559, 2559, 2560, 2560, 2561, 2561, 2562, 2562,
2750, 2998
translatable strings
locatable 974
translate and learn property
use 976
Translated Text 983, 983, 984
translation prefix
display 974
tree picker
examples 154
tree picker attribute
add 154
trigger condition
create 594
example 596
for assessments 363
trigger conditions 596
trigger rules 1799
trigger rules upgrade 1799
triggered inbound actions
untrusted users 2841
© 2017 ServiceNow. All rights reserved. 4064
Troubleshoot Performance
- cache https 1999
- compression 1999
- response time on forms 1996
- transaction log response times 1995

Troubleshoot Performance traceroute
- network response times 1998
- ping times 1998

tutorial 3375

Twilio
- account associating 2969
- endpoints configuring 2971
- Legacy notify configuring 2969
- type ahead suggestions, updating 1080

U
- ui actions 679
  - UI actions
    - visibility by role 738
    - visibility with views 738
- UI configuration
  - favicon 277
- UI macros 3873
- UI page
  - use as gauge 513
- UI pages 3871, 3872
- UI Pages 3869
- UI policy
  - converting to a data policy 942
  - creating from a data policy 943
- UI routing action
  - create 3991
- UI scripts
  - create 3913
  - run 3914
- UI15
  - configure roles to switch to UI16 15
  - switch to UI16 11
- UI15 compare style 12
- UI16
  - activating 10
  - configure roles to switch to UI15 15
  - switch to UI15 11
- UI16 compare style 12
- unencrypted attachments
  - encrypt with script 2537
- Unencrypted Single Sign-On
  - ASP script 4019
- unique index 1634
- unique record identifier 1632
- unsupported data sources
  - for JDBC driver 1523
- untranslated string
  - export 975
- untrusted users
  - preventing 2841
  - triggered inbound actions 2841

Upgrade History 1969, 1971
- upgrade monitor
  - upgrade platform 1974
  - upgrade system 1974
- Upgrade Monitor /
  - upgrade instance 1974
- upgrades 1969
- URL
  - navigating 53, 54, 55
  - URL endpoints 3873
- URL module
  - custom filter URL 53
    - opens in new window 52
- URL query
  - execute script 3877
- Usage Analytics 2530
- Usage Overview 1766

Use A BSM Map
- access the map 2212
  - CI filters by group 2220
  - create a relationship 2224
  - delete a relationship 2226
  - direction and level 2219
  - edit a relationship 2225
  - export a map 2222
  - filter the map 2217
  - manage map relationships 2223
  - manage the map 2213
  - tasks by type and date 2220
  - view a change to a relationship 2225
  - view a relationship 2223
  - view map information glyphs 2221

Use Chat
- add a favorite room 2328
- add favorite user 2317
- change the display 2328
  - chat window 2324
  - create a chat room 2319, 2321
  - delete a chat room 2306
  - invite a user 2328
  - join chat room 2323, 2324, 2324
  - join one-to-one chat 2323
  - message 2326
  - remove a favorite room 2329
  - remove favorite user 2317
  - send emoticons 2326
  - send message 2325
  - set chat preferences 2329
  - start a one-to-one chat 2319
  - task 2321, 2324
  - task record 2324
  - update profile 2312
  - update status 2313
  - use favorites list 2315
  - view a room 2318
  - view online user 2317

Use Content Blocks
- CMS block tag 2073
  - create a content block 2072
  - detailed content block 2072
  - iframe 2098
use feeds in live feed
activity stream 2381
create a group feed 2393
create record feeds 2394
delete a feed 2398
group feed 2396
invite a new member 2396, 2398
leave a feed 2398
live feed 2381, 2398
mark a feed as a favorite 2395
navigate 2398
other modules 2398
participate in a feed 2394
record feed 2396
record feeds 2381
records 2381
remove a member from a feed 2397
subscribe to a feed 2398
view a feed 2396
view a feed member 2396
use hashtags in live feed
add an image to a hashtag 2375
bookmark a hashtag 2377
change a hashtag 2374
follow a hashtag 2376
merge hashtags 2374
posted message 2378
remove a hashtag 2378
tag messages 2372
unsubscribe a hashtag 2376
view available hashtags 2373
Use HTML Fields
editing functions 800
default before-query business rule 4017
extended functions 801
highlight text in the TinyMCE editor 803
insert a line break 803
style a table 803
TinyMCE HTML editor 792
TinyMCE version 3 and htmlArea editor 798
TinyMCE version 4 editor 796
use HTML source mode in the htmlArea editor 803
use the editors 792
use interactive SQL
ODBC 3499
use knowledge
tables installed with 194
use teams in live feed
automatic team creation 2417
create a team 2415
delete a team 2423
dependency 2422
e-mail 2391, 2421
e-mail notification 2420
feed notification 2421
invite a new member 2418
join a team 2417
leave a team 2421
manage 2421
membership request 2421
modify a team 2421
remove a member 2419
reply to notifications 2391, 2421
subscribe 2420, 2421
team 2421
team notification 2420
view a team feed 2418
view a team member 2418
Use the Smartphone Interface
tutorials 1176
Use Time Zones
service level agreements 1057
set a system time zone 1056
time zone 1057
Use versioning
Scripted REST APIs 3231
useful attachment scripts 2782
useful scripts
Modify a GlideDateTime Field Value 4002
Restrict Form Views by Role 240
useful approval assignment scripts 4006
Useful Scripts
abort a database action in a business rule 4016
default before-query business rule 4017
scheduling scripts 4014
useful field scripts 4010
workflow use case 4021, 4023, 4024, 4025, 4026
User administration
company 1747
user creation
enabling automatic creation 2840
user guide
activating for upgraded instances 194
configuring the help page 199
creating a help page 194
creating a section 195
creating entry points for display 200
creating help documents 196
introduction to 192
tables installed with 194
user interface
application navigator 44
navigation filter 44
UI11 40, 41
UI11 perspectives 42
UI15 38, 39
Using approval rules

Using Access Control Rules

Users and Groups

user specific UI differences

user skill

User sessions

user registration

user roles

User Session Inheritance API

user registration

approvers related list

generate approvals

using forms

form headers

UI11 form headers

UI15 form headers

UI16 form headers

Using Forms

use HTML fields

use feeds in live feed

use hashtags in live feed

use feeds in Live Feed

use teams in Live Feed

use chat

use teams in live feed

use hashtags in live feed

use feeds in Live Feed

record feeds

post content in Live Feed

pop-up forms

hierarchical lists

pipes 468

User Interface

generate approvals

impersonate user

debug

acl

impersonate a user

associate to a group

assign roles

allow to view a profile

assign roles

associate to a group

create

creating from incoming email

Users and Groups

impersonate a user

associate to a group

create

Using Access Control Rules

acl

debug

impersonate user

using approval rules

generate approvals

© 2017 ServiceNow. All rights reserved. 4067
Utility Activities
join activity 3551

V
validation script
deactivate 1747
variables
adding 642
Verizon
ebonding
migrating 1699
integration 1699
videos
removing channels in workflow editor 3534
view
bubble charts 379
survey results 527
view canceled transactions 2037
view legacy chat messages
journal field 2306
View Management
administering views 236
create a view rule 238, 239
creating view 237
delete view 238
switch views 236
view rules
configuring 1232
View System Logs
audited tables 1058, 2562
available logs 1057
configuration logging 1058, 2559
customer updates table 1058, 2561
e-mail logs 1059, 2555, 2750
event logs 1061, 2558, 2998
import logs 1062, 2558
log history 1062, 2560
log utilities 1063, 2560
logged information 1062
Outbound email notification recipients 1063, 2557
system diagnostics application 1064, 2561
system logs 1064, 2552
system mailboxes 1065, 2562
transaction logs 1065, 2553
workflow logging 1066, 2559
viewing
attachment

table 749
calls
conference 2966
conference call 2966
decision
matrix 445
Legacy notify
properties 2973
Legacy Notify
conversation 2965
message 2964
question 2962
matrix
decision 445
properties
Legacy Notify 2973
table
attachment 749
visual task boards
activate 2236
board structure 2232
browser support 2232
card details 2242
checklists
configuring 2250
creating from templates 2249
creating new 2248
deleting 2249
configuring 2258
connect
sharing visual task boards in 2258, 2258, 2281, 2288
deleting 2239
flexible boards
adding tasks to 2242
configuring card limit 2261
creating 2239
freeform boards
adding personal tasks to 2240
adding tasks to 2240, 2241
configuring card limit 2261
creating 2238
moving cards between boards 2251
tasks 2240
guided boards
adding tasks to 2242
configuring card limit 2261
creating 2239
installed components
business rules 2237
client scripts 2237
notifications 2238
properties 2236
tables 2236
labels
configuring 2260
lanes
adding 2240
deleting 2240
modifying 2240
list view 2253
members
adding 2251
removing 2251
SLA indicators 2256
task cards
archiving 2250
assigning 2244
labels 2243
moving between boards 2251
moving between lanes 2251
opening form view of 2251
types 2232
using 2238
viewing board information 2252
Visual task boards
installed components
script includes 2237
VPN communication
create address 2545
VPN connection
alternatives 2252
redundant tunnels 2545
ServiceNow and business network 2545
VPN service
request 2548

W
watch lists
configure 838
configure email notifications 838
configure multiple 838
use 837
Web Proxy
basic proxy setup 1433
ntlm authentication 1434
proxy servers 1435
soap clients 1435
Web Service
security 3403
Web Service
JSON v2 3380, 3381, 3384, 3385, 3385, 3387, 3389
web service API
aggregate 3261
API functions 3256
Data Modification API 3264, 3265, 3267, 3270, 3271, 3272
Data Retrieval API 3256, 3257, 3258, 3260, 3261
deleteMultiple 3272
deleteRecord 3271
get 3260
getKeys 3257
getRecords 3258
insert 3265
insertMultiple 3267
SOAP API 3256
update 3270
Web service import sets 1583, 1587, 1590
Web Service Import Sets
Debugging 1587
eamples 1591
Security requirements 1591
web service security 2459, 2463, 2464
web services
ACL 3283
basic authentication 3283, 3284
internal integration communications 3286
internal integration users 3286
mutual authentication 3298
role requirements 3283
security 3282, 3283, 3284, 3286
SOAP envelope header 3291
SOAP Security Policies 3293
strict security 3297, 3297
UsernameToken 3290
WS Security profile 3287
WS-security 3286, 3286
WS-Security 3291
WS-security profile 3287
X.509 token 3289
Web services
Available web services 3044
Web Services
outbound soap web service 3434, 3435, 3435, 3437, 3439, 3440, 3441, 3442, 3449, 3449, 3449, 3449, 3450
Web Services APIs
table API Java examples 3108
Web Services C Sharp .NET End to End Tutorial
call web service 3372
configuration 3372
results 3376
source code 3375
troubleshoot 3377
use a service reference 3374
use a web reference 3375
visual studio .net 3372
Web Services-Security (WSS) 3295
website
HTML fields
linking 807
linking
HTML fields 807
weight for metric categories 324
welcome page
content 241
sections 243
welcome page content
choose language 241
company-specific 242
widgets
condition count 859
Windows 2662
Wizard Panels
change a banner setting 659
create a fixed wizard banner step 660
field setters 663
use wizard banners 664
wizard script
examples 3899
record generators 3899
wizard scripts 3899
Wizard UI policy 3915,
3915, 3916
wizard variables 642
wizards
transitions 641
variables 643
Wizards
activate a system wizard 664
advanced customization 666
advanced wizard 666, 666, 667, 668, 668, 668, 669,
670, 670, 671, 671
catalog order and checkout panels 659
concepts 665
create a basic wizard 636
create a panel 660

© 2017 ServiceNow. All rights reserved. 4069
create a survey wizard 653, 653, 654, 655, 656, 656, 657, 658
create the first panel 638
create the second panel option 639
define a basic transition 664
define a wizard variable 637
edit a panel 662
KB viewers 663
panels 659, 659, 660, 663, 664
public 646, 646, 646, 647, 647, 648, 648
publish the wizard 644
record generators 663
redirect panel 663
survey wizards
configure the survey property 649
create 650
create a dynamic effect 650
create a page 649
create a survey condition 650
create the email notification 650
define a question 651
define a transition 650
Installed components 651
plugin 652
test 653
test the wizard 640
words 1094
workflow
canceling 3646
canceling
context
administering 3643
menu options 3531
performance timing 3675, 3675
roles 3642
scheduling 3649
title bar controls 3531
Workflow
Documentation tab 3544
estimated run time 3670
estimated run time updates 3671
Estimated Runtime tab 3544
execution history table 3674
execution path timelines 3671
gauges 3673
General tab 3544
History (Past Contexts) tab 3544
Inputs tab 3544
monitor 3672
outlying run times 3670
overview 3523, 3642
properties 3544
run time metrics, enable 3670
run time metrics, overview 3670
Schedule tab 3544
Stages tab 3544
troubleshoot 3671
variables 3563, 3563, 3563, 3564, 3565, 3565, 3565, 3565, 3566, 3566, 3566, 3566, 3567, 3570
Version tab 3544
Workflow Operations Dashboard 3672
workflow activities
activity designer 3676
activity level pinning 3650
activity result value 3761
adding to a workflow 3755
core 3676
correcting skipped activities 3763
creating custom transitions 3759
duplicating in a workflow 3759
elements listed 3754
locked versions 3768
managing conditions 3759
managing transitions between 3757
manual update of 3652
Orchestration 3676
out of date, indicator of 3651
pinning introduction 3650
publishing a custom activity 3768
setting maximum allowed 3657
updated, indicator of 3651
utility activities 3734, 3734, 3735, 3735, 3738, 3738, 3738, 3738, 3739, 3739, 3740, 3740, 3740, 3742, 3743
viewing descriptions of 3753
workflow level pinning 3650
Workflow activities
Workflow activities overview 3676
Workflow Activities
multiple timer activities in one workflow 3768
Notify 2913, 3710
SLA Percentage Timer 3721
timer 3722
timer activities 3721
workflow activity
Forward to notify client 2922, 3715
queue 2923, 3719
Send SMS 2916, 3718
Workflow activity
Call 2915, 3716
Forward call 2917, 3711
Hang up 2919, 3713
Join conference call 2914, 3717
Reject call 2921, 3714
Say 2921, 3715
Workflow Activity
Input 2918, 3711
Play 2920, 3713
Record 2920, 3714
Workflow Administration
error 3669
errors 3664, 3665, 3665, 3668
workflow timelines 3658, 3658, 3662, 3663
Workflow Concepts
data separation 3637
delegated administration 3635
domain separation 3635
exit conditions 3524
restrictions 3634
transitions 3524
updates 3588
workflow activities 3523
workflow caching 3641
workflow editing 3525
workflow engine operation order 3639
workflow permissions 3639
invoking subflows in 3557
palette 3530
passing variables to a subflow 3556
prepare for adding subflow 3561
preparing a subflow for 3557
publish 3551
selecting activities for 3530
subflow inputs 3556
subflows in 3552
user interface, introduction to 3526

WS-security
error logs 3296
error messages 3296
WS-Security
security policy 3294
WSDL Schema
hierarchical data 3301

X

XML files
export 1495
import 1495
XML Files
exporting 1494, 1494, 1495
importing 1494, 1494, 1495
XML import sets 1605
XML web service
parameters 3393
XMLContent
getIterator() 2687
getIterator(String xPath) 2688
XMLContent API 2687
XMLDocument Script Object 3981, 3983
XMLElement
encodedQueryFor(String tableName) 2693
getAttributeValue(String attribute) 2694
getIterator(String xPath) 2691
getIteratorOverAllChildren() 2692
getName() 2694
valueFor(String tableName, String fieldName) 2693
XMLElement API 2691
XMLElementIterator
hasNext() 2689
next() 2690
XMLElementIterator API 2689

Z

Zing
junk filter, disabling 1077
text search, debugging 1077
Zing text search 1067